COURANT COMPUTER SCIENCE PUBLICATIONS

COURANT COMPUTER SCIENCE NOTES

A101	ABRAHAMS, P.	*The PL/I Programming Language*, 1979, 151 p.
C66	COCKE, J. & SCHWARTZ, J.	*Programming Languages and Their Compilers*, 1970, 767 p.
D86	DAVIS, M.	*Computability*, 1974, 248 p.
M72	MANACHER, G.	*STPL: A Low-Level Language in the Style of Algol*, 1971, 496 p.
M81	MULLISH, H. & GOLDSTEIN, M.	*A SETLB Primer*, 1973, 201 p.
S91	SCHWARTZ, J.	*On Programming: An Interim Report on the SETL Project. Generalities, The SETL Language and Examples of Its Use*, 1975, 675 p.
S99	SHAW, P.	*GYVE--A Programming Language for Protection and Control in a Concurrent Processing Environment*, 1978, 668 p.
S100	SHAW, P.	" *Vol. 2*, 1979, 600 p.
W78	WHITEHEAD, E. G., Jr.	*Combinatorial Algorithms*, 1973, 104 p.

COURANT COMPUTER SCIENCE REPORTS

No. 1	WARREN, H., Jr.	*ASL: A Proposed Variant of SETL*, 1973, 326 p.
2	HOBBS, J. R.	*A Metalanguage for Expressing Grammatical Restrictions in Nodal Spans Parsing of Natural Language*, 1974, 266 p.
3	TENENBAUM, A.	*Type Determination for Very High Level Languages*, 1974, 171 p.
4	OWENS, P.	*A Comprehensive Survey of Parsing Algorithms for Programming Languages*, Forthcoming, 652 plus pages.
5	GEWIRTZ, W.	*Investigations in the Theory of Descriptive Complexity*, 1974, 60 p.
6	MARKSTEIN, P.	*Operating System Specification Using Very High Level Dictions*, 1975, 152 p.
7	GRISHMAN, R.(Ed.)	*Directions in Artificial Intelligence: Natural Language Processing*, 1975, 107 p.
8	GRISHMAN, R.	*A Survey of Syntactic Analysis Procedures for Natural Language*, 1975, 94 p.
9	WEIMAN, CARL	*Scene Analysis: A Survey*, 1975, 62 p.
10	RUBIN, N.	*A Hierarchical Technique for Mechanical Theorem Proving and Its Application to Programming Language Design*, 1975, 172 p.
11	HOBBS, J. R. & ROSENSCHEIN, S. J.	*Making Computational Sense of Montague's Intensional Logic*, 1977, 41 p.
12	DAVIS, M & SCHWARTZ, J.T.	*Correct-Program Technology/Extensibility of Verifiers*, with *Appendix* by E. Deak, 1977, 146 p.
13	SEMENIUK, C.	*Groups with Solvable Word Problems*, 1979, 77 p.

NOTES: Available from Department IN. Prices on request.
REPORTS: Available from Ms. Lenora Greene.
No. 1,3,4,6,7,8,10 available in xerox copy only .. (out-of-print).

COURANT INSTITUTE OF MATHEMATICAL SCIENCES
251 Mercer Street
New York, N. Y. 10012

COURANT INSTITUTE OF MATHEMATICAL SCIENCES

Computer Science NSO-13

GROUPS WITH SOLVABLE WORD PROBLEMS

Christine Semeniuk

Report No. NSO-13 prepared under
Grant No. NSF-MCS 7600116 from
the National Science Foundation

TABLE OF CONTENTS

ABSTRACT

This paper considers two aspects of the word problem for groups: first, some similarities between derivations from the relations of a group and proofs from a set of axioms in logic, and second, the computational difficulty of word problems and the problem of constructing groups with solvable word problems of some preassigned degree of difficulty.

Given a presentation of a group in terms of generators and relations, we define two words on the generators to be equal if and only if one can be transformed into the other in a finite number of steps using the relations of the group. Defining equality in a group as derivability from a set of axioms suggests formulating the word problem for a group as the derivability problem for a formal system. The system we choose is equational logic. We show that there exist some striking analogies between results from logic and results about groups: nontrivial groups correspond to consistent systems, groups with solvable word problem to decidable systems, and simple groups to complete systems. This suggests that results about formal systems could be used to obtain results about decidability in groups.

A group has word problem in level $\mathcal{E}^n$ of the Grzegorczyk hierarchy if the running time of the algorithm solving the word problem is in $\mathcal{E}^n$. Groups having word problem solvable in levels $\mathcal{E}^n$ $(n \geq 2)$ of the Grzegorczyk hierarchy are given. The groups are constructed using a standard procedure of constructing a group given the presentation of a semigroup. Semigroups are constructed following J. Robinson's method of functional equations: the semigroups are decidable systems of functional equations which define functions in $\mathcal{E}^n$. This technique for constructing semigroups is particularly elegant, for it has the advantage of exhibiting the semigroup as a concrete system. We show that if the semigroup has word problem in $\mathcal{E}^n$ (and not lower), then the resulting group has word problem in $\mathcal{E}^n$ (and not lower). Hence there exist groups with arbitrarily difficult, but solvable word problems.

CHAPTER 1

INTRODUCTION

Given a presentation of a group (or a semigroup) G, we define two words w_1, w_2 on the generators of G to be equal if and only if one can be transformed into the other in a finite number of steps using the relations of G. The word problem is the problem of deciding whether two arbitrary words on the generators of G are equal, or, alternatively, if there exists a proof of "$w_1 = w_2$" from the relations of G.

Equating truth in G with derivability from a set of axioms suggests the possibility of formulating the word problem for G as the derivability problem for some type of formal system. The first part of the paper is concerned with this. Once this is done, we can consider the questions usually asked about formal systems, such as consistency, decidability, and completeness, and use results about particular formal systems to obtain results about groups and semigroups corresponding to these systems. In this analogy, nontrivial groups correspond to consistent systems, groups with solvable word problem to decidable systems, and simple groups to complete systems. The last analogy is particularly striking, for, strangely enough, although it has been known for a long time that complete recursively axiomatized theories are decidable, it was only recently noted that recursively presented simple groups have solvable word problems.

The major part of the paper is concerned with the problem of exhibiting finitely presented groups with word problems solvable in levels $\mathcal{E}^n$ of the Grzegorczyk hierarchy.

A group (or semigroup) has word problem in $\mathcal{E}^n$ if the running time of the algorithm solving the word problem is in $\mathcal{E}^n$. The group used to prove recursive unsolvability of the word problem [Rotman, 1973] is constructed from a presentation of a semigroup computing a nonrecursive function. We use the same technique of obtaining a group presentation from a semigroup presentation and show that if the semigroup has word problem in $\mathcal{E}^n$ $(n \geq 4)$, then the resulting group has word problem in $\mathcal{E}^n$.

Groups with word problems in $\mathcal{E}^2$ and $\mathcal{E}^3$ are exhibited. Semigroups are constructed following Julia Robinson's method of functional equations [J. Robinson, 1968]. This method has the advantage of exhibiting a semigroup (or a group) as a concrete structure, namely, as a semigroup (or a group) of transformations of certain structures. This approach has been first used by MacKenzie and Thompson [1973], and then considerably improved by Higman [1974] in exhibiting an infinite family of finitely presented infinite simple groups.

The problem of exhibiting finitely presented groups with word problem in $\mathcal{E}^n$ has been looked at by Cannonito and Gatterdam [1973] (and [Gatterdam, 1973]). They approached it indirectly by showing that the Higman construction of embedding a recursively presented group in a finitely presented group preserves $\mathcal{E}^n$-decidability, and by exhibiting a recursively presented $\mathcal{E}^n$-decidable group. Their notion of $\mathcal{E}^n$-decidability is different from ours; they consider a group to have a "realization" in the natural numbers, and the $\mathcal{E}^n$-decidability of a group is defined relative to a fixed indexing of the group.

CHAPTER 2

GROUPS AND SEMIGROUPS AS DEDUCTIVE SYSTEMS

Simple groups behave very much like complete theories. This chapter is an attempt to explain why.

A result due to Birkhoff states that a semigroup with identity, i.e., a monoid, is isomorphic to the semigroup of all endomorphisms of some unary algebra. We can thus view the generators of a semigroup as functions, and the relations as relations between functions. Derivations using the semigroup relations consist of formal substitutions; no reference is made to the domain over which functions are interpreted, and there exists the possibility that there are several different interpretations.

A deductive system of the type suggested above, i.e., one in which the only admissible formulas are equations and the only rule of inference is substitution (of equals for equals or terms for variables) is called an *equational system*. In Section 2.1 we present A. Tarski's formulation of equational logic. We show that with each presentation of a semigroup we can associate an equational system, and that derivations in the semigroup correspond to proofs within the system. This provides a link between logic and semigroup theory, for now we can use results about equational theories to obtain results about semigroups.

In Section 2.3 we examine the concepts of consistency and completeness applied to equational theories arising from semigroup presentations. These are too general for the class of equations we will be concerned with, and new definitions of consistency and completeness are given, relativized to certain sets of equations. We then show that there are some interesting analogies between results from logic and from group and semigroup theory.

2.1 Equational Systems — Definitions and Main Results

An *algebra* is a system $\mathcal{A} = \langle A, \mathcal{O}_1, \ldots, \mathcal{O}_k \rangle$ where A is a nonempty set and $\mathcal{O}_1, \ldots, \mathcal{O}_k$ is a finite sequence of finitary operations on A. A is called the *universe* and $\mathcal{O}_1, \ldots, \mathcal{O}_k$ the *fundamental operations* of $\mathcal{A}$. The sequence of ranks of $\mathcal{O}_1, \ldots, \mathcal{O}_k$ is called the *type* of $\mathcal{A}$. A fundamental operation of rank 0 is identified with the unique element which constitutes its range; such an element is called a *distinguished element* of the algebra. If all the operations have rank one, then $\mathcal{A}$ is called a *unary* algebra.

Below are some examples of algebras; to avoid parentheses we will put an operation symbol in front of the variables to which it refers.

(i) Algebras $\langle A, \cdot \rangle$ with one binary operation satisfying

$$\cdot\cdot x\ y\ z = \cdot x \cdot y\ z \quad \text{for all } x,y,z \text{ in } A$$

All such algebras are called *semigroups*.

(ii) A group is an algebra $\langle G, \cdot, {}^{-1}, 1 \rangle$ with one binary operation $\cdot$, one nullary operation 1, and one unary operation ${}^{-1}$ satisfying:

(1) $\cdot\cdot\ x\ y\ z = \cdot x \cdot y\ z$

(2) $\cdot 1\ x = \cdot x\ 1 = x$

(3) $\cdot x\ x^{-1} = \cdot x^{-1}\ x = 1$

for all elements x,y,z in G.

(iii) Let $\mathcal{O}_1, \ldots, \mathcal{O}_k$ be a sequence of operations and let $X = \{x_i \mid i \in I\}$. The set of all possible terms that can be built up from X and $\mathcal{O}_1, \ldots, \mathcal{O}_k$ is an algebra. For if $\mathcal{O}$ is an n-ary operation and $w_1, \ldots, w_n$ are terms, then the term $\mathcal{O}w_1 \ldots w_n$ is taken to be the result of applying $\mathcal{O}$ to $w_1, \ldots, w_n$. The algebra so constructed is called a *free algebra* and X is called a *free generating set*.

Many properties of algebras can be expressed by means of equations, i.e. formulas of the form $\alpha = \beta$ where α and β are terms of the algebra being discussed. The study of abstract algebras and the equations which hold in them was initiated by G. Birkhoff [1935]. He established the rudiments of a formal system, now known as *equational logic*, in which equations are the only admitted formulas. There have been more recent formulations of equational logic since then. Eilenberg and Wright [1967] gave a formulation in terms of category theory. We present here A. Tarski's [1968] formulation of equational logic as part of predicate logic: equations "$\alpha = \beta$" with variables $x_1,\dots,x_n$ are treated as if they were sentences $\forall x_1,\dots,x_n$ $(\alpha = \beta)$.

The underlying language L consists of an infinite number of variables, a finite number of function symbols, and =, the symbol for equality. Terms and formulas are defined as in first order logic. An *equation* is a formula of the form $\alpha = \beta$, where α and β are terms. The equation "x = x" is always included among the axioms, and the rules of inference are

(i) substituting terms for variables, and

(ii) replacing equals by equals.

Let $\sigma = <\sigma_1,\dots,\sigma_k>$ be the sequence of operation symbols and Σ a set of equations over these symbols. The models of Σ are algebras $<A,\mathcal{O}_1,\dots,\mathcal{O}_k>$ such that for each i, $\mathcal{O}_i$ has the same rank as σ_i , and which satisfy Σ. We say that $\alpha = \beta$ is a *logical consequence* of Σ if $\alpha = \beta$ is valid in every model of Σ, and $\alpha = \beta$ is *derivable* from Σ if there exists a proof of $\alpha = \beta$. The completeness theorem for equational logic, proved by Birkhoff [1935] states that the notions of logical consequence and derivability are equivalent.

By an *equational theory* θ we mean a set of equations closed under logical consequence. A class K of algebras is called an *equational class* if K is the class of all models of some set Σ of equations. We say that K is the equational class defined by Σ.

Birkhoff [1935] has given a purely algebraic characterization of equational classes: a class K is equational if and only if all subalgebras, direct products, and homomorphic images of algebras in K are in K. We note here that Eilenberg and Schutzenberger [1976] have studied what happens to this theorem if one restricts one's attention to finite algebras only.

Contrary to the case in first order logic, where there exist sets of sentences with no models, every set of equations Σ always has models -- the free algebras, obtained from free algebras of terms by equating terms using the equations of Σ. These may be trivial, consisting of only one element. The free algebras satisfying Σ can be thought of as the "most general" algebras satisfying Σ, for every algebra generated by m elements and satisfying Σ is a homomorphic image of the free algebra on m generators [Birkhoff, 1935].

2.2 Associating an Equational System with a Semigroup

Let S be a finitely generated semigroup with generators $s_1,\dots,s_k$ and relations $\alpha_i = \beta_i$, $i \in I$, an index set, denoted $S = (s_1,\dots,s_k|\alpha_i = \beta_i,\ i \in I)$. Let α and β be words on the generators of S. We say that $\alpha = \beta$ is *derivable* from S if there exists a sequence of words $\gamma_1,\dots,\gamma_n$ such that $\gamma_1 = \alpha$, $\gamma_n = \beta$ and γ_{i+1} is obtained from γ_i by using a relation of S, and we write $S \vdash \alpha = \beta$ if $\alpha = \beta$ is derivable from the relations of S. Consider the equational system with function symbols $s_1,\dots,s_k$ and axiom set $\Sigma_S = \{\alpha_i(x) = \beta_i(x) | i \in I\}$. We write $\Sigma_S \vdash \alpha(x) = \beta(x)$ if $\alpha(x) = \beta(x)$ is provable from Σ_S.

Theorem 2.2.1 Let α,β be any two words on the generators of S. Then $S \vdash \alpha = \beta \Leftrightarrow \Sigma_S \vdash \alpha(x) = \beta(x)$.

Proof: ($\Rightarrow$) Suppose $S \vdash \alpha = \beta$.

Then there exists a sequence of words $\gamma_1, \ldots, \gamma_m$, where γ_1 is α, γ_m is β and γ_{i+1} is obtained from γ_i using a relation in S. Thus, $\gamma_1 = \gamma_2 = \ldots = \gamma_m$. The following sequence is a proof of $\alpha(x) = \beta(x)$ from Σ_S:

$$\Sigma_S \vdash x = x$$

$$\Sigma_S \vdash \alpha(x) = \alpha(x) \qquad \text{(Substituting term for variable)}$$

$$\left.\begin{array}{l} \Sigma_S \vdash \alpha(x) = \gamma_2(x) \\ \Sigma_S \vdash \alpha(x) = \gamma_3(x) \\ \qquad\vdots \\ \Sigma_S \vdash \alpha(x) = \gamma_{m-1}(x) \\ \Sigma_S \vdash \alpha(x) = \beta(x) \end{array}\right\} \text{(Substituting equals for equals)}$$

($\Leftarrow$) Suppose $\Sigma_S \vdash \alpha(x) = \beta(x)$. To show $S \vdash \alpha = \beta$, we use induction on the length of the proof from Σ_S. If $\alpha(x) = \beta(x)$ is an axiom (by assumption, it is not "$x = x$"), then $S \vdash \alpha = \beta$.

Suppose $\alpha(x) = \beta(x)$ is obtained from $\alpha'(x) = \beta'(x)$ by substituting a term $\gamma(x)$ for a variable x, and $S \vdash \alpha' = \beta'$.

Then $\Sigma_S \vdash \alpha'\gamma(x) = \beta'\gamma(x)$.

Since $S \vdash \alpha' = \beta'$, $S \vdash \alpha'\gamma = \beta'\gamma$.

Suppose $\alpha(x) = \beta(x)$ is obtained from $\alpha'(x) = \beta'(x)$ by replacing equals by equals. Without loss of generality, we can assume that the equality was one of the axioms. Then, it is easily seen that $S \vdash \alpha = \alpha'$ and $S \vdash \beta' = \beta$. By hypothesis, $S \vdash \alpha' = \beta'$, and putting these proofs together, we have $S \vdash \alpha = \beta$. □

Remark 2.2.2 Note that in Σ_S there are no equations of the form $\alpha(x) = x$. However, if S contains an identity e, we can consider the system Σ_S with the additional equation $e(x) = x$ and the above theorem still holds.

This theorem sets up the correspondence between derivations in semigroup S and proofs from Σ_S. Whenever we speak of a set of equations Σ_S, we will assume that the underlying equational system contains functions for all the generators of S (for it is possible that not all the generators appear in the set of defining relations).

For the rest of this section let Σ be a set of equations between unary functions containing the *same variable* on both right and left-hand sides. We wish to look at the models of Σ.

First of all, the free algebras are models of Σ.

Proposition 2.2.3 Σ always has a nontrivial model, i.e., a model consisting of more than one element.

Proof: Consider the free algebra on n generators ($n \geq 2$) for Σ together with the equations $g(x) = x$ for g a function symbol in Σ. This algebra contains exactly n elements. □

Proposition 2.2.4 Σ contains models of every cardinality.

Proof: If all the functions are interpreted to be the identity, then we can take an arbitrary set to be a model of Σ. □

Remark 2.2.5 In Chapter 4, we will present systems of functional equations where the intended domain of interpretation is N, the natural numbers.

Remark 2.2.6 Let G be a group; suppose that a presentation for G contains the relations $gg^{-1} = e$ for g a generator of G. Let Σ_G be the equational system associated with G and suppose that Σ_G contains the equation $e(x) = x$. By Cayley's theorem, G itself is a model of Σ_G: we can think of G as a group of permutations (unary functions), and the relations of G as relations between functions.

From Propositions 2.2.3 and 2.2.4, we see that a set of equations Σ containing $g(x) = x$ for all functions g has nontrivial models. Such interpretations are always possible, but they are rather uninteresting and it is reasonable to consider these models as in some sense "trivial". It is interesting to note here that a group is defined to be trivial if $w = e$ for all elements w; regarding the elements as permutations, this means that the only possible permutation (consistent with the relations) is the identity permutation. This further suggests that we consider those models of Σ "trivial" in which all functions are interpreted to be the identity. This is done in the next section.

2.3 Consistency, Completeness, Decidability - Restricted Definitions

Definition 2.3.1 (Tarski [1968]) An equational theory θ is called *consistent* if it does not contain all equations (with given operation symbols) or, equivalently, if it does not contain the equation $x = y$.

It immediately follows that θ is consistent if and only if θ has a nontrivial model.

Definition 2.3.2 (Tarski [1968]) An equational theory θ is called *complete* if there is no consistent theory which properly includes θ. Or, equivalently, θ is complete if for all equations $\alpha = \beta$ with given operation symbols, either $\theta \vdash \alpha = \beta$ or $\theta \cup \{\alpha = \beta\}$ is inconsistent.

Remark 2.3.3 The following theorems from predicate logic extend to equational theories:

(i) (Lindenbaum's theorem) Every consistent theory has a consistent, complete extension.

(ii) Complete, recursively axiomatized theories are decidable.

We will now apply these concepts to unary equational theories containing only equations involving one variable (for the theories arising from semigroup presentations are of this type). Let θ be such a theory.

Proposition 2.3.4 θ is consistent.

Proof: $x = y$, being an equation in two variables, cannot be a logical consequence of equations of one variable. □

We remark that in Proposition 2.2.3 we showed that θ always has a nontrivial model, giving another proof that θ is consistent.

Proposition 2.3.5 θ is complete $\Leftrightarrow$ θ contains all equations involving one variable.

Proof: ($\Rightarrow$) Suppose not all equations of one variable are contained in θ, e.g. $\alpha(x) = \beta(x) \notin \theta$. Then by 2.3.4, neither is $\theta \cup \{\alpha(x) = \beta(x)\}$ inconsistent, i.e., θ is not complete.

($\Leftarrow$) Suppose θ contains all equations of one variable. Let $\alpha(x) = \beta(y)$ be an arbitrary equation. Then $\alpha(x) = x$ and $\beta(y) = y$ are in θ by assumption. Therefore, $\theta \cup \{\alpha(x) = \beta(y)\} \vdash x = y$, i.e., $\theta \cup \{\alpha(x) = \beta(y)\}$ is inconsistent. Therefore θ is complete. □

Thus we cannot make distinctions between different theories θ based on consistency and completeness. This happens because the set of equations relative to which consistency is defined is in a sense "too large"; it includes equations $\alpha(x) = \beta(y)$ (two variables), and for the theories θ, no equations involving two variables can ever be proved. This suggests restricting the set of "all equations" to "all equations of one variable", and then defining consistency and completeness relative to this set of equations. To emphasize the fact that only equations of one variable are involved, we call these notions 1-consistency and 1-completeness. Let θ be, as above, a unary equational theory containing only equations of one variable.

Definition 2.3.6 θ is 1-*consistent* if it does not contain all equations of one variable (with given operation symbols) or, equivalently, if it does not contain the equations $f(x) = x$ for all functions f.

It follows from the completeness theorem for equational logic that θ is 1-consistent if and only if there exists a model of θ in which not all functions are the identity.

A model can be thought of as being trivial if it is a model of an "inconsistent" set of equations. The "trivial" models in this sense are the ones in which all functions are the identity. Thus, a system of equations may have models which are nontrivial in the general sense, i.e., consisting of more than one element, and trivial in this sense.

<u>Definition 2.3.7</u> θ is 1-*complete* if there is no 1-consistent theory which properly includes θ, or, equivalently, if for all equations $\alpha = \beta$ of one variable, either $\alpha = \beta \in \theta$ or $\theta \cup \{\alpha = \beta\}$ is 1-inconsistent.

<u>Definition 2.3.8</u> θ is 1-*decidable* if all equations of one variable are decidable.

Note that if θ is 1-decidable, then θ is in fact decidable, for no equation $\alpha(x) = \beta(y)$ involving two variables is provable.

The corresponding analogues of theorems given in 2.3.3 hold with the restricted notions of consistency and completeness:

<u>Proposition 2.3.9</u> A 1-complete recursively axiomatized theory θ is decidable.

<u>Proof</u>: Let $\alpha = \beta$ be an arbitrary equation.

(i) If $\alpha = \beta$ is an equation of two variables, then $\alpha = \beta$ is not provable.

(ii) If $\alpha = \beta$ is an equation of one variable, then either $\theta \vdash \alpha = \beta$ or $\theta \cup \{\alpha = \beta\}$ is 1-inconsistent, and 1-inconsistency can be checked using 2.3.6. □

<u>Proposition 2.3.10</u> (Lindenbaum's theorem) A 1-consistent theory θ has a 1-consistent, 1-complete extension.

Proof: Given θ, let $A = \{\Gamma \mid \Gamma$ is a set of equations containing only one variable and $\theta \cup \Gamma$ is 1-consistent$\}$. A is inductive, i.e., every chain has an upper bound. By Zorn's lemma, A has a maximal element, Δ. $\theta \cup \Delta$ is 1-complete. □

2.4 Logic, Groups, and Semigroups

This section connects ideas from logic with semigroups and groups. Throughout this section, S will denote a semigroup with identity, G a group, and θ_S and θ_G the equational theories associated with S and G, respectively. We assume that S and G are finitely generated and presentations are given, and that θ_S and θ_G contain the equation $e(x) = x$, where e is the identity. The semigroup (or group) consisting of only the identity element is called *trivial*.

The following follows from definitions and 2.2.1:

Proposition 2.4.1 (i) θ_S is 1-consistent $\Leftrightarrow$ S is nontrivial. (ii) S has solvable word problem $\Leftrightarrow$ θ_S is decidable.

Let w_1 and w_2 be words of S. S^{w_1,w_2} is the semigroup whose whose relations are those of S together with $w_1 = w_2$.

Definition 2.4.2 S is *simple* if for any two words w_1, w_2 such that $w_1 \neq w_2$, S^{w_1,w_2} is trivial (i.e., S is a semigroup with only two congruences: the identity, modulo which it remains unchanged, and the universal congruence, modulo which it becomes trivial. This is also called *congruence simplicity*).

Corollary 2.4.3 Let $g_1, g_2, \ldots, g_n$ be the generators of S. Then S is simple if for any two words w_1, w_2 with $w_1 = w_2$ the following holds in S^{w_1,w_2} : $g_1 = g_2 = \ldots = g_n = e$.

Proposition 2.4.4 S is simple $\Leftrightarrow$ θ_S is 1-complete.

Proof: S is simple $\Leftrightarrow$ for all words w_1, w_2 in S, either $S \vdash w_1 = w_2$ or $S \cup \{w_1 = w_2\}$ is trivial.

θ_S is 1-complete $\Leftrightarrow$ for all equations $\alpha = \beta$ involving one variable either $\theta_S \vdash \alpha = \beta$ or $\theta_S \cup \{\alpha = \beta\}$ is 1-inconsistent.

If either of α or β is "x" (suppose α is "x"), then $\theta_S \vdash x = \beta \Leftrightarrow \theta_S \vdash e(x) = \beta$. Thus, it suffices to consider equations $\alpha = \beta$ in which neither α nor β is "x". The result follows from 2.2.1. □

Proposition 2.4.5 Finitely generated recursively presented simple semigroups have solvable word problem.

Proof: Let S be a finitely generated recursively presented simple semigroup (with identity). By 2.4.4 θ_S is 1-complete. θ_S is recursively axiomatized, and so by 2.3.9 θ_S is decidable. By 2.2.1, a decision procedure for θ_S gives a decision procedure for S. □

Let G be a finitely generated group. A *maximal normal subgroup* is a subgroup N of G such that G/N is simple. A presentation of G/N consists of the relations of G together with $w = e$ for $w \in N$.

Proposition 2.4.6 Every finitely generated group has a maximal normal subgroup.

Proof: In view of the above remarks, the proposition says that any finitely generated group can be made simple by adding relations. But by 2.2.1 this is Proposition 2.3.10 for θ_S. □

Simple groups, then, do act like complete theories, and enumerating a maximal normal subgroup is analogous to completing a theory. We note that for first order theories, if we start with a consistent decidable theory, then we can *effectively* obtain a consistent complete extension. The proof depends on the fact that for T a first order theory and $\{A_1,\ldots,A_n\}$ a finite set of first order sentences,

$T \cup \{A_1,\ldots,A_n\}$ is decidable (for any first order sentence B, $T \cup \{A_1,\ldots,A_n\} \vdash B \Leftrightarrow T \vdash A_1 \ \& \ \ldots \ \& \ A_n \to B$). This does not carry over to equational theories. It can be seen by observing that for a group G, G has solvable word problem does *not* imply that $G \cup A$ has solvable word problem, for A an arbitrary finite set of relations on the generators of G.

2.5 Finite Theories and Residually Finite Groups

Residually finite groups behave similarly to finite equational theories in the following sense: if G is a residually finite group, w a word on the generators of G, and $w \neq 1$ in G, then $w \neq 1$ in some finite homomorphic image of G; if θ is a finite equational theory, $\alpha = \beta$ an equation involving the function symbols of θ, and $\alpha \neq \beta$ in θ, then there is a finite model of θ in which $\alpha \neq \beta$. Now, any homomorphic image of G satisfies the relations of G. So, in both cases an equation can be disproved by considering finite models. This is further explored in this section.

Definition 2.5.1 For an equational theory θ, $\underline{F\theta}$ is the equational theory whose valid equations are just those in the notation of θ which hold in every finite model of θ.

An equational theory θ such that $\theta = F\theta$ is called a *finite equational theory*. Note that $F\theta$ is consistent if and only if θ has a nontrivial finite model. (These definitions are restrictions to equational theories of the ones for first order theories given in Trahtenbrot [1950]).

Clearly logical validity implies finite validity. To show that the converse does not hold, we first need the following definition.

Definition 2.5.2 Let C be a set of algebras. K is *generated by* C if every member of K is obtained by taking direct products, homomorphic images, and subalgebras of algebras in C.

Note that by Birkhoff's theorem [1935] K is an equational class. Therefore there exists a set of equations Σ such that K is the set of all models of Σ.

To show that finite validity does not imply logical validity consider the class K generated by an infinite cyclic group and a finite cyclic group of order k. Clearly, every finite algebra in K satisfies $x \cdot x \cdot \ldots \cdot x = 1$ (k times), but this equation is not logically valid.

Now suppose that K is generated by a set C of finite algebras. Then Th(K), the set of equations satisfied by every algebra in K is a finite theory. (In fact, Th(K) = Th(C).) For every algebra in K is obtained from algebras in C via certain operations and if $\alpha = \beta$ is true in every finite algebra in K, then it is true in every algebra in C, and therefore true in every algebra obtained from algebras in C by these operations. Furthermore, if C contains only a finite number of algebras, then Th(C) is effectively enumerable, for we can effectively verify if an equation holds for all algebras of C.

<u>Proposition 2.5.3</u> If $\theta = F\theta$ and θ is finitely axiomatizable, then θ is decidable.

<u>Proof</u>: If $\theta = F\theta$, then for any equation $\alpha = \beta$, either $\theta \vdash \alpha = \beta$ or there exists a finite model in which θ is false. Suppose θ is finitely axiomatized by the set of equations Σ. Then we can effectively enumerate everything provable from Σ. Because Σ is finite, we can enumerate all finite algebras, check if they are models of Σ, check if $\alpha = \beta$ holds, and so enumerate the set of equations which fail to hold in some finite model. Since $\theta = F\theta$, every equation $\alpha = \beta$ must appear on one of these two lists. □

We can get equational theories as in the proposition by picking for the set C above a finite algebra such that the set of equations valid in C is finitely axiomatizable. A particular example is a finite group (Oates and Powell [1964]).

Definition 2.5.4 A group G is *residually finite* if for each $w \neq 1$ in G, there exists a normal subgroup N such that G/N is finite and $w \neq 1$ in G/N.

Proposition 2.5.5 Let G be a residually finite group. Then θ_G is a finite equational theory.

Proof: By 2.2.6, G is a model of θ_G. Let $\alpha(x) = \beta(x)$ be an equation on the function symbols of θ_G. By 2.2.1 $\theta_G \vdash \alpha(x) = \beta(x) \Leftrightarrow G \vdash \alpha = \beta$. If $G \nvdash \alpha = \beta$, then there exists a finite homomorphic image of G in which $\alpha \neq \beta$. But this homomorphic image is also a model of θ_G. Hence there exists a finite model of θ_G in which $\alpha(x) \neq \beta(x)$ and so θ_G is a finite equational theory. □

Proposition 2.5.6 Finitely presented residually finite groups have solvable word problem.

Proof: By 2.5.5 the equational theory θ_G associated with a residually finite group G is a finite theory. By 2.5.3 θ_G is decidable. □

CHAPTER 3

COMPUTATIONAL COMPLEXITY OF WORD PROBLEMS

The Grzegorczyk hierarchy of functions, defined by A. Grzegorczyk [1953] is composed of an infinite sequence of classes $\mathcal{E}^0 \subset \mathcal{E}^1 \subset \mathcal{E}^2 \subset \ldots$, each properly contained in the next and having as union the class of all primitive recursive functions. Each class can be inductively defined as the smallest class of functions containing certain initial functions and closed under the operations of substitution and bounded primitive recursion. Using the limited form of recursion has the effect of restricting the growth of the functions within each class: a function obtained by recursion from functions in $\mathcal{E}^n$ can be in the class $\mathcal{E}^n$ only if it is bounded by a function already in the class. A new function, dominating all functions in $\mathcal{E}^n$ is obtained by unbounded recursion from functions in $\mathcal{E}^n$. This function is used as an initial function for $\mathcal{E}^{n+1}$.

The Grzegorczyk hierarchy is particularly interesting because it relates the computational difficulty of a function with its location in the hierarchy. A function can be shown to lie in $\mathcal{E}^n$ $(n \geq 3)$ by showing that the amount of space or the amount of time needed to compute it is a function in $\mathcal{E}^n$. This criterion is machine independent in the sense that the amount of space or the amount of time can be measured relative to a large number of different computing machines and the result is the same, showing that the computational difficulty of a function is a quality intrinsic to the function, not depending on characteristics of specific machines.

Other work in categorizing functions with respect to computational difficulty has been done by Cobham [1964] and Ritchie [1963]. Ritchie's hierarchy, based on space requirements, is a refinement of the class of elementary functions.

We say that a group G has word problem in $\mathcal{E}^n$ if there exists a presentation Π_G for G and an algorithm solving the word problem whose computational complexity is in $\mathcal{E}^n$, i.e., the amount of time or the amount of space used by the algorithm is an $\mathcal{E}^n$-function of the length of the input. We show that for finitely generated groups, having word problem in $\mathcal{E}^n$ is a property of the group, and not of the particular presentation of the group. The word problem for free groups is shown to be in $\mathcal{E}^2$, and the direct product of G_1 and G_2 is shown to have word problem in $\mathcal{E}^n$ whenever both G_1 and G_2 have word problems in $\mathcal{E}^n$.

3.1 The Grzegorczyk Hierarchy - Definitions and Main Results

Let C be a class of functions from N to N (N = nonnegative integers).

Definition 3.1.1 C is *closed under substitutions* if C is closed under the following three operations:

(i) Composition of functions. If $f(x_1,\ldots,x_{k-1},x_k,x_{k+1},\ldots x_n)$ and $g(y_1,\ldots,y_m)$ are in C, then so is their composition $f(x_1,\ldots,x_{k-1},g(y_1,\ldots,y_m),x_{k+1},\ldots,x_n)$.

(ii) Identification of variables. If $f(x_1,\ldots,x_j,\ldots x_k,\ldots x_n)$ is in C, then so is the function obtained by replacing both the variables x_j and x_k by y: $f(x_1,\ldots y,\ldots y,\ldots x_n)$. Here y is a variable different from all the x_i's.

(iii) Substitution of a constant. If $f(x_1,\ldots,x_k,\ldots,x_n)$ is in C, then so is the function $f(x_1,\ldots,0,\ldots,x_n)$ obtained by substituting the constant 0 for x_k.

Definition 3.1.2 Let g, h, and j be given functions and let f be the function obtained from them as follows:

$$f(\underline{u},0) = g(\underline{u}) \qquad (\underline{u} = (u_1,\ldots,u_n))$$

$$f(\underline{u},x+1) = h(\underline{u},x,f(\underline{u},x))$$

$$f(x,\underline{u}) \leq j(x,\underline{u})$$

Then f is said to be defined from g, h, and j by *bounded primitive recursion.* The class C is *closed under bounded primitive recursion* if whenever g, h, and j are in C, then so is the function f as given above.

Definition 3.1.3 C_n is the class of functions of n arguments which belong to the class C.

Definition 3.1.4 The function $F(x_1,\ldots,x_n,t)$ of n+1 arguments is a *universal function* for the class C_n provided that for each function $f(x_1,\ldots,x_n)$ of n arguments the function f belongs to C_n if and only if there exists a number t such that for each $x_1,\ldots,x_n$, $f(x_1,\ldots,x_n) = F(x_1,\ldots,x_n,t)$.

To define the Grzegorczyk hierarchy, we first present the following functions:

$$f_0(x,y) = y + 1$$

$$f_1(x,y) = x + y$$

$$f_2(x,y) = (x+1)(y+1)$$

For $n \geq 2$,

$$f_{n+1}(0,y) = f_n(y+1,\ y+1)$$

$$f_{n+1}(x+1,y) = f_{n+1}(x, f_{n+1}(x,y))$$

Definition 3.1.5 For $n \in N$, the class $\mathcal{E}^n$ (Grzegorczyk's n^{th} class) is the smallest class of functions C such that

(i) the functions x+1, $U_1(x,y) = x$, $U_2(x,y) = y$, and $f_n(x,y)$ are in C.

(ii) C is closed under substitutions and bounded primitive recursion.

Another definition of Grzegorczyk's hierarchy, using a different set of initial functions and the operation of limited exponentiation is given in Meyer and Ritchie [1967].

We note now some results about the Grzegorczyk hierarchy which will be used later.

Theorem 3.1.6 (Grzegorczyk [1953])

(i) $\mathcal{E}^n \subsetneqq \mathcal{E}^{n+1}$

(ii) $\bigcup_{n\geq 0} \mathcal{E}^n = P$ (the primitive recursive functions)

(iii) $\mathcal{E}^3$ is the class of elementary functions

(iv) For $n \geq 3$, the class $\mathcal{E}^{n+1}$ contains the universal function for the class $\mathcal{E}^n_1$

(v) The function $f_{n+1}(x,x)$ increases faster than any function of the class $\mathcal{E}^n_1$

(vi) Every recursively enumerable set is enumerated by a function of class $\mathcal{E}^0$

(vii) For every function $f \in \mathcal{E}^n$ $(n \geq 2)$ there exists k such that $f(x) < f_{n+1}(k,x)$ for all x

(viii) $f_{n+1}(x,y) > y$ for $n \geq 1$

(ix) $f_{n+1}(x+1,y) > f_{n+1}(x,y)$ for $n \geq 0$

(x) $f_n(x,y+1) > f_n(x,y)$ for $n \geq 0$.

Let T be a deterministic Turing machine with a single two-way infinite tape, a finite alphabet, and a finite number of states. T can execute the following instructions: erase a symbol and print another one in its place, move one tape square to the left, or move one tape square to the right. T computes a numerical function f(n) by starting out with n encoded on its tape as a sequence of n+1 1's (assuming 1 is in the alphabet of T). The computation terminates when T reaches a final state and the value of f(n) is the number of 1's on the tape. We can define two functions:

$t_T(n)$ = the number of steps (instruction executions) in the computation of T, starting with n encoded on its tape

$s_T(n)$ = the number of tape squares scanned during the course of computation.

Theorem 3.1.7 (Cobham [1964]) For each $k \geq 3$, the following five statements are equivalent:

(i) $f \in \mathcal{E}^k$

(ii) There exists a Turing machine T which computes f and such that $t_T \in \mathcal{E}^k$

(iii) There exists a Turing machine T which computes f and a function $g \in \mathcal{E}^k$ such that for all n, $t_T(n) \leq g(n)$

(iv)-(v) same as (ii)-(iii) with s_T in place of t_T.

This theorem is also true for wider classes of computing machines; it holds for Turing machines with more than one tape or with multidimensional tapes. It also holds if the set of possible instructions is extended to include, for example, erasure of an entire tape. This suggests that the computational difficulty of a function is an intrinsic property of the function.

Definition 3.1.8 The *characteristic function* of a set S is a function χ_S such that

$$\chi_S(x) = \begin{cases} 0 \text{ if } x \in S \\ 1 \text{ if } x \notin S \end{cases}$$

We say that a set S is in $\mathcal{E}^n$ if and only if its characteristic function χ_S is. Note that if S is in $\mathcal{E}^n$, then so is its complement $\bar{S}$, for $\chi_{\bar{S}} = 1 \dot{-} \chi_S$.

It is not clear from the way the Grzegorczyk hierarchy is defined whether there exist 0-1 valued functions in $\mathcal{E}^{n+1}-\mathcal{E}^n$, i.e., whether there exist sets in $\mathcal{E}^{n+1}-\mathcal{E}^n$. Requiring functions defined by primitive recursion to be bounded by a function already in the class has the effect of restricting the growth of the functions. Thus the growth of a function seems to be a major factor in determining where it lies in the Grzegorczyk hierarchy. However, we have the following result.

Theorem 3.1.9 For $n \geq 3$, there exist 0-1 valued functions in $\mathcal{E}^{n+1} - \mathcal{E}^{n}$.

Proof: For $n \geq 3$, let ψ_{n+1} be the universal function for the class $\mathcal{E}_1^n$ (cf. 3.1.6 (iv)). Define

$$g_{n+1}(x) = \begin{cases} 1 & \text{if } \psi_{n+1}(x,x) \text{ even} \\ 0 & \text{if } \psi_{n+1}(x,x) \text{ odd} \end{cases}$$

Clearly, $g_{n+1} \in \mathcal{E}^{n+1}$. If $g_{n+1} \in \mathcal{E}^{n}$, then there exists x_0 such that $g_{n+1}(x) = \psi_{n+1}(x_0,x)$. Then

$$g_{n+1}(x_0) = \psi_{n+1}(x_0,x_0) = \begin{cases} 1 & \text{if } \psi_{n+1}(x_0,x_0) \text{ even} \\ 0 & \text{if } \psi_{n+1}(x_0,x_0) \text{ odd} \end{cases}$$

which cannot happen. Hence g_{n+1} cannot be in $\mathcal{E}^{n}$. □

$S_{n+1} = \{x \mid g_{n+1}(x) = 0\}$ is a set in $\mathcal{E}^{n+1} - \mathcal{E}^{n}$.
It is not known whether this result also holds for $n < 3$.

3.2 Iteration and the Grzegorczyk Hierarchy

Definition 3.2.1 Let h_1 and h_2 be given functions and let g be the function obtained from them as follows:

(i) $$g(0,y) = h_1(y)$$

(ii) $$g(x+1,y) = h_2 g(x,y)$$

Then g is said to be defined from h_1 and h_2 by *iteration*. If in addition to (i) and (ii), $g(x,y) \leq h_3(x,y)$ where h_3 is a given function, then g is said to be defined from h_1, h_2 and h_3 by *bounded iteration*.

Let $j(x,y)$ be a one-to-one mapping of the set of all pairs of natural numbers onto the set of natural numbers. Let k and ℓ be the inverse functions: $kj(x,y) = x$ and $\ell j(x,y) = y$. Any such function j is called a *pairing function*. Examples of pairing functions are:

(i) (Cantor pairing function) $j(x,y) = ((x+y+1)(x+y))/2 + x$

(ii) (Exponential pairing function) $j(x,y) = (2x + 1)2^y - 1$ (J. Robinson [1967]).

Note that any pairing function j has to be at least quadratic, i.e., $\mathcal{E}^2$ is the lowest level of the Grzegorczyk hierarchy in which a pairing function can occur.

Theorem 3.2.2 If $f,h \in \mathcal{E}^n$ $(n \geq 2)$, then the function g defined by

$$g(0,y) = h(y)$$

$$g(x+1,y) = fg(x,y)$$

is in $\mathcal{E}^{n+1}$.

Proof: By 3.1.6 (vii), $f \in \mathcal{E}^n \Rightarrow$ there exists k_1 such that

$$f(y) < f_{n+1}(k_1,y)$$

Also by 3.1.6 (vii), $h \in \mathcal{E}^n \Rightarrow$ there exists k_2 such that

$$h(y) < f_{n+1}(k_2,y)$$

Let $k = \max\{k_1, k_2\}$. Then

$$f(y) < f_{n+1}(k,y)$$

and

$$h(y) < f_{n+1}(k,y) \quad .$$

We will show by induction on x that

$$g(x,y) < f_{n+1}(k+x,y) \qquad (*)$$

For $x = 0$, $g(0,y) = h(y) < f_{n+1}(k,y)$.
Suppose the inequality (*) holds for x. Then

$$\begin{aligned} g(x+1,y) = fg(x,y) &< f_{n+1}(k,g(x,y)) < f_{n+1}(k,f_{n+1}(k+x,y)) \\ &< f_{n+1}(k+x,f_{n+1}(k+x,y)) = f_{n+1}(k+x+1,y). \end{aligned}$$

□

Remark 3.2.3 By using R. Robinson's technique for simplifying recursion [1947], we can show that if j,k and ℓ are added to the initial functions of $\mathcal{E}^n$ ($n \geq$ level of j), then bounded recursion can be replaced by bounded iteration of the form

$$\begin{aligned} g(0,y) &= h_1(y) \\ g(x+1,y) &= h_2 g(x,y) \\ g(x,y) &\leq i(x,y) \end{aligned}$$

in the definition of the Grzegorczyk hierarchy.

We will now show that for $n \geq 2$, the functions f_{n+1} can be obtained by iteration from functions in $\mathcal{E}^n$. For $n \geq 2$, we define

$$\begin{aligned} f^*_{n+1}(0,y) &= y \\ f^*_{n+1}(x+1,y) &= f_n(f^*_{n+1}(x,y)+1, f^*_{n+1}(x,y)+1) \end{aligned}$$

Lemma 3.2.4 For $n \geq 2$, $f^*_{n+1}(x+1,y) = f^*_{n+1}(1,f^*_{n+1}(x,y))$.

Proof: $f^*_{n+1}(x+1,y) = f_n(f^*_{n+1}(x,y)+1, f^*_{n+1}(x,y)+1)$ and

$$\begin{aligned} f^*_{n+1}(1,f^*_{n+1}(x,y)) &= f_n(f^*_{n+1}(0,f^*_{n+1}(x,y)) +1,\ f^*_{n+1}(0,f^*_{n+1}(x,y))+1) \\ &= f_n(f^*_{n+1}(x,y)+1, f^*_{n+1}(x,y)+1) \end{aligned}$$

□

Lemma 3.2.5 For $n \geq 2$, $f^*_{n+1}(1,f^*_{n+1}(x,y)) = f^*_{n+1}(x,f^*_{n+1}(1,y))$.

Proof: By induction on x. For $x = 0$,
$f^*_{n+1}(1,f^*_{n+1}(0,y)) = f^*_{n+1}(1,y) = f^*_{n+1}(0,f^*_{n+1}(1,y))$.
Suppose the lemma holds for any x. Then

$$f^*_{n+1}(1,f^*_{n+1}(x+1,y)) \overset{3.2.4}{=} f^*_{n+1}(1,f^*_{n+1}(1,f^*_{n+1}(x,y)))$$

$$\overset{\text{hyp}}{=} f^*_{n+1}(1,f^*_{n+1}(x,f^*_{n+1}(1,y))) \overset{3.2.4}{=} f^*_{n+1}(x+1,f^*_{n+1}(1,y)). \quad \square$$

Lemma 3.2.6 For $x \geq 2$, $f^*_{n+1}(x_1+x_2,y) = f^*_{n+1}(x_1,f^*_{n+1}(x_2,y))$.

Proof: By induction on x_2. If $x_2 = 0$, there is nothing to prove. If $x_2 = 1$, this is Lemma 3.2.4. Suppose the lemma holds for $x_2 > 1$, for fixed x_1. Then,

$$f^*_{n+1}(x_1+x_2+1,y) \overset{\text{hyp } \& \text{ } 3.2.4}{=} f^*_{n+1}(1,f^*_{n+1}(x_1,f^*_{n+1}(x_2,y)))$$

$$\overset{3.2.5}{=} f^*_{n+1}(x_1,f^*_{n+1}(1,f^*_{n+1}(x_2,y))) \overset{3.2.4}{=} f^*_{n+1}(x_1,f^*_{n+1}(x_2+1,y))$$

$\square$

Theorem 3.2.7 For $n \geq 2$, $f_{n+1}(x,y) = f^*_{n+1}(2^x,y)$.

Proof: By induction on x. For $x = 0$:

$$f^*_{n+1}(2^0,y) = f^*_{n+1}(1,y) = f_n(f^*_{n+1}(0,y)+1,f^*_{n+1}(0,y)+1)$$

$$= f_n(y+1,y+1) = f_{n+1}(0,y)$$

Suppose the theorem holds for x. Then

$$f^*_{n+1}(2^{x+1},y) = f^*_{n+1}(2\cdot 2^x,y) = f^*_{n+1}(2^x+2^x,y)$$

$$\overset{3.2.6}{=} f^*_{n+1}(2^x,f^*_{n+1}(2^x,y)) \overset{\text{hyp}}{=} f_{n+1}(x,f_{n+1}(x,y))$$

$$= f_{n+1}(x+1,y) \quad \square$$

Corollary 3.2.8 For $n \geq 2$, $f^*_{n+1}(x,y) \in \mathcal{E}^{n+1} - \mathcal{E}^n$.

Proof: For $n = 2$, it can be shown by induction on x that for all y,

$$f^*_{n+1}(x,y) > y^x \in \mathcal{E}^3 - \mathcal{E}^2 .$$

For $n > 2$, if f^*_{n+1} were in $\mathcal{E}^n$, then so would $f^*_{n+1}(2^x,y) = f_{n+1}(x,y)$, since $\mathcal{E}^n$ is closed under composition. But this implies that $f_{n+1} \in \mathcal{E}^n$ -- contradiction. Thus,

$$f^*_{n+1}(x,y) \in \mathcal{E}^{n+1} - \mathcal{E}^n . \qquad \square$$

Now, the function $g(x) = 2^x$ can easily be defined from the functions $s(x) = x+1$ and $f^*_3(x,y)$ by bounded iteration:

Let $g_1(x)$ be defined by:

$$g_1(0) = 0$$

$$g_1(x+1) = ssg_1(x)$$

$$g_1(x) \leq f^*_3(1,x) \qquad \text{(i.e., } g_1(x) = 2x\text{)}$$

Let $g(x)$ be defined by:

$$g(0) = 1$$

$$g(x+1) = g_1 g(x)$$

$$g(x) \leq f^*_3(x,2). \quad \text{Then, } g(x) = 2^x.$$

Hence, for $n \geq 3$, f^*_n can replace f_n in the definition of $\mathcal{E}^n$, and together with 3.2.3 we have:

Proposition 3.2.9 For $n \geq 3$, $\mathcal{E}^n$ can be defined as the smallest class of functions containing $s(x) = x+1$, $U_1(x,y) = x$, $U_2(x,y) = y$, $f^*_n(x,y)$, j, k, ℓ and closed under the operations of substitution and bounded iteration.

Proposition 3.2.10 The functions $f_1(x,y) = x+y$ and $f_2(x,y) = (x+1)(y+1)$ can be defined from $s(x) = x+1$, j, k, ℓ by substitution and iteration.

Proof: $f_1(x,y)$ can be defined by

$$f_1(0,y) = y$$

$$f_1(x+1,y) = 1+f_1(x,y)$$

To define $f_2(x,y)$, first define f_2':

$$f_2'(0,y) = j(0,y+1)$$

$$f_2'(x+1,y) = j(sk,f_1(k,\ell))f'(x,y)$$

Then,

$$f_2(x,y) = \ell f_2'(x,y) \ . \qquad \square$$

From 3.2.9 and 3.2.10 we see that every function in $\mathcal{E}^n$ ($n \geq 3$) can be defined in a special way:

Proposition 3.2.11 Any function in $\mathcal{E}^n$ ($n \geq 3$) can be defined from $s(x) = x+1$, $U_1(x,y) = x$, $U_2(x,y) = y$, j, k, and ℓ by repeatedly constructing a new function from previously obtained functions by substitutions and iteration. Furthermore, unbounded iteration is needed only to define the initial function f_n^*.

3.3 $\mathcal{E}^n$-Decidable Groups

Definition 3.3.1 A group G *has word problem in* $\mathcal{E}^n$ if there exists a presentation Π_G for G, a Turing machine T and a function $g \in \mathcal{E}^n$ such that for any word w on the generators of G, T decides in $\leq g(\text{length}(w))$ steps whether or not $w = 1$ in G. A similar definition holds for semigroups.

Note that for $n \geq 3$, this is equivalent to using "space" in place of "time" (3.1.7).

The word problem for groups is treated similarly to the recognizability problem for formal languages: the set of words of G which equal 1 is the set of finite strings of symbols accepted by a Turing machine (Hopcroft and Ullman [1969]).

Other analogies between word problems for groups and recognizability of languages have been exhibited. G. Sacerdote [1977] defines a finitely generated group to be *automatic* if the set of words of the group which equal 1 is the set of finite strings of symbols accepted by a finite automaton. It has been shown that automatic groups are precisely the finite groups. An algebraic characterization for context free groups is asked for.

We say that G is $\mathcal{E}^n$-*decidable* if G has word problem in $\mathcal{E}^n$.

Given a group G (and a presentation Π_G) and an element $w \in G$, we can try to see if w = 1 in G by using the relations of G to transform w into 1. If w is in fact equal to the identity, then a proof will be found. If G is a finitely presented group then we can easily construct a (nondeterministic) Turing machine to simulate this procedure. Roughly, the alphabet of the Turing machine would contain the generators of G and it would work by locating the right or left side of a relation and replacing it. To estimate the number of instructions executed in applying a relation to w, we just note that to do so, a portion of w might have to be moved, depending on the length of the word being substituted. The number of instructions executed in applying one relation is about k·length(w) and the length of the resulting word is $\leq$ k·length(w), where k is a constant equal to the number of instructions needed to move one letter of w. By induction the number of instructions executed in applying n relations is $\leq n \cdot k^n \cdot \text{length}(w)$.

If we have a bound on the number of relations that need to be used to see if w = 1, then the procedure is effective. Suppose that we need $\leq$ f(length(w)) relations to decide if w = 1. Then we can construct a Turing machine which works by first computing f(length(w)) and then like the one described above, with the extra step that it subtracts 1 from f(length(w)) when a relation is used. It terminates when f(length(w)) becomes 0 or w gets transformed into 1.

We can also construct a determinnstic Turing machine, not much more complex with respect to time and space requirements than the nondeterministic one above, to solve the word problem for G. W. Savitch [1970] has shown that a nondeterministic Turing machine using s(n) tape squares on an input of length n $(s(n) > \log_2(n))$ can be simulated by a deterministic Turing machine using at most $(s(n))^2$ tape squares. It can easily be seen that

<u>Remark 3.3.2</u> For $n \geq 3$, if the number of relations used in deciding $w = 1$ is $\leq h(\text{length}(w))$, $h \in \mathcal{E}^n$, then G has word problem in $\mathcal{E}^n$.

Here we are taking into consideration that the length of the word may increase in applying a relation. If this does not happen, then this result holds for $n \geq 2$.

In what follows we try to locate where the word problem of a group lies with respect to the Grzegorczyk hierarchy. The number of relations used to decide if $w = 1$ (for a finitely presented group) is a convenient way of measuring the difficulty of the word problem. We remark that in counting the number of relations used, we consider the following to be relations: $1 \cdot g_i = g_i$, $g_i \cdot 1 = g_i$, $g_i g_i^{-1} = 1$, $g_i^{-1} g_i = 1$, where g_i is a generator of G.

<u>Theorem 3.3.3</u> A finitely generated free group has word problem in $\mathcal{E}^2$.

<u>Proof</u>: Let G be a free group on n generators. It is easily seen that the word problem for G is equivalent to the membership problem for the context free language consisting of all well-formed expressions over n pairs of matching left and right parentheses, where parentheses may balance on either side. It has been shown [Hopcroft and Ullman, 1969] that the number of steps needed to recognize a context free language is in $\mathcal{E}^2$. □

This result can also be proved by noting that the number of relations needed to show $w = 1$ in a free group is $\leq$ length(w) and using 3.2.2. (The length of the word never increases in the course of a derivation.)

Let $G_1 \subseteq G_2$, $G_1 = (g_1,\ldots,g_n|R_1)$, $G_2 = (g_1',\ldots,g_m'|R_2)$. Since G_1 is embedded in G_2 , for each generator g_i of G_1 there exists a word w_{g_i} on the generators of G_2 (under the embedding isomorphism). For v a word in G_1 , let w_v be the word of G_2 obtained by replacing g_i with w_{g_i} and g_i^{-1} with $w_{g_i}^{-1}$. Thus, we have that $v = 1$ in $G_1 \Leftrightarrow w_v = 1$ in G_2. Now if the length of the longest w_{g_i} is k, then the length of w_v is at most k·length(v). We can conclude the following:

(i) If G_2 has word problem in $\mathcal{E}^n$ $(n \geq 1)$, then G_1 has word problem in $\mathcal{E}^n$ (if the number of steps required is g(length(u)) for u a word of G_2 , then the number of steps for a word v in G_1 is g(k·length(v))).

(ii) If a finitely generated group G has word problem in $\mathcal{E}^n$ $(n \geq 1)$ for some finite system of generators, then G has word problem in $\mathcal{E}^n$ for any other finite system of generators.

(iii) The word problem for a finitely generated subgroup of G cannot be more difficult than the word problem for G.

Given two groups A, B with word problems in $\mathcal{E}^n$, it is frequently interesting to see how difficult is the word problem for the group G obtained from A and B by using some standard construction from group theory. We show that the word problem for the direct product of A and B is not more difficult (i.e., higher in the Grzegorczyk hierarchy) than the more difficult of A and B. We will show in Chapter 5 that the Britton extension of a group A can have word problem substantially harder than A.

Definition 3.3.4 Let $K = (k_1,\ldots,k_n|R)$ and $Q = (q_1,\ldots,q_m|S)$. Let θ be a homomorphism $\theta\colon Q \to \mathrm{Aut}(k)$ defined by $\theta_x(k) = xkx^{-1}$ for all $k \in K$, $x \in Q$. The *semidirect product* $K \times_\theta Q$ is the set of all ordered pairs $(k,x) \in K \times Q$ under the binary operation $(k_1,x)(k_2,y) = (k_1\theta_x(k_2),xy)$.

A presentation for $K \times_\theta Q$ is the following:

generators: $k_1,\ldots,k_n,\ q_1,\ldots,q_m$

relations: $R,\ S$

$$(*) \qquad q_ik_j = w_{ij}q_i \ , \quad q_ik_j^{-1} = w_{ij}^{-1}q_i \ , \qquad i=1,\ldots,m$$

$$(**) \qquad q_i^{-1}k_{jj} = \underline{w}_{ij}q_i^{-1}, \quad q_i^{-1}k_j^{-1} = \underline{w}_{ij}^{-1}q_i^{-1}, \qquad j=1,\ldots,n$$

(w_{ij} and $\underline{w}_{ij}$ are words on $k_1,\ldots,k_n$).

(Actually, since θ_{q_i} is an automorphism, the w_{ij} are a set of generators for K and relations (**) can be obtained from (*).)

We note the following facts:

(i) Every element of $K \times_\theta Q$ can be expressed as kq, where $k \in K$, $q \in Q$.

(ii) Let $g = kq$, where $k \in K$, $q \in Q$. Then $g = 1$ in $K \times_\theta Q$ if and only if $k = 1$ and $q = 1$.

Proposition 3.3.5 If K and Q have word problem in $\mathcal{E}^n$, $n \geq 3$, then $K \times_\theta Q$ has word problem in $\mathcal{E}^n$.

Proof: To decide if $w = 1$, we put w into normal form (i) and then use the algorithms for solving the word problem for K and Q (by (ii)).

Let r be the maximum length of any w_{ij} or $\underline{w}_{ij}$. To put w into normal form, scan from left to right, moving the q's to the right. Let $|w|$ = length of w. The length of the word can increase by r each time a relation (*) or (**) is used; thus, to move one q_i over to the right can require at most $|w|$ relations and the length of the resulting word is $\leq r|w|$. The next q_i has to be moved across a word of length at most $r|w|$, resulting in a word of length at most $r^2|w|$. By induction, to move all the q's to the right requires at most $|w|^2r^{|w|-1}$ relations and the length of the resulting word is $r^{|w|}|w|$.

Let K have word problem solvable in $\leq g(|w|)$ steps, $g \in \mathcal{E}^n$. Let Q have word problem solvable in $\leq h(|w|)$ steps, $h \in \mathcal{E}^n$. Then $K \times_\theta Q$ has word problem solvable in

$$\leq f(|w|) = |w|^2 r^{|w|} + g(|w| r^{|w|}) + h(|w| r^{|w|}) \text{ steps,}$$

and f is in $\mathcal{E}^n$ for $n \geq 3$. □

Remark 3.3.6 If $r = 1$ in the above proof, then the proposition holds for $n \geq 2$.

CHAPTER 4

$\mathcal{E}^n$-DECIDABLE SEMIGROUPS

If Σ is a system of functional equations with function symbols $f_1,\dots,f_n$, then we can associate with Σ a semigroup S_Σ , consisting of words over $f_1,\dots,f_n$ with the operation of juxtaposition, and whose set of relations is Σ. This provides a very elegant method for constructing semigroups, for we have a concrete interpretation of the generators as functions. J. Robinson [1968] showed that the word problem for semigroups is recursively unsolvable by using a system of functional equations defining a nonrecursive function. We use this idea to construct decidable semigroups.

However, defining recursive functions does not guarantee decidability for the system Σ. We give simple examples of equations which we would expect to be derivable from Σ, but which are not. The reason for this is that we think of the functions in Σ as being defined over the natural numbers, but in defining derivability from Σ, we are considering Σ to be a formal system without reference to any domain of interpretation. Thus, systems of functional equations are equational systems (cf. Chapter 2) and there exist domains for the functions other than the natural numbers. Some examples are given.

A system of functional equations Σ, however, can be made decidable by adding relations to obtain a new system Σ' in such a way that the functions computed by Σ' are the same as those computed by Σ. We show that for $n \geq 4$, there exist systems of functional equations Σ computing functions in $\mathcal{E}^n$ and such that Σ is $\mathcal{E}^n$-decidable.

4.1 Functional Equations - Definitions and Properties

In this section we define and give the main properties of systems of functional equations, as presented in J. Robinson [1968].

Small letters m,n,t,x,y will refer to elements of N, all others to functions from N to N.

Definition 4.1.1 The function f is obtained by *general recursion* from functions a,b,g,h if

(i) $fa = g$, $fb = hf$, and

(ii) every natural number n belongs to $\text{Range}(b^m a)$ for some $m \geq 0$.

From (i) we have $f\,b^m a(t) = h^m g(t)$ for any $t \in N$. If a,b,g, and h are computable and (ii) holds, then given any n we can effectively find m and t, so that $n = b^m a(t)$ and then f is effectively computable. In particular, if $a = o$ (the zero function $o(x) = 0$) and $b = s$ (the successor function $s(x) = x+1$), then (i) and (ii) are satisfied.

Throughout this chapter, s will denote the successor function and o will denote the zero function.

J. Robinson [1968] showed that

Theorem 4.1.2 Every recursive function of one variable can be obtained from o and s by repeated compositions and general recursions from previously defined functions.

Thus, every primitive recursive function and therefore every function in $\mathcal{E}^n$ can be so defined. It is not immediately clear that if f is in $\mathcal{E}^n$ $(n \geq 0)$ and obtained as above, whether all auxiliary functions obtained in the process are also in $\mathcal{E}^n$.

We will show that for $n \geq 3$, any function in $\mathcal{E}^n$ can be obtained as in the theorem in such a way that all functions obtained on the way are in $\mathcal{E}^n$. The proof is essentially the same as that given in J. Robinson [1968], with verifications that the functions concerned are in $\mathcal{E}^n$.

Let j be a pairing function. Let n_j be the Grzegorczyk level of j, i.e., $j \in \mathcal{E}^{n_j}$.

Definition 4.1.3 f is defined from a, b, h, g, and p by *bounded general recursion* if f is obtained by general recursion from a, b, g, h and $f(x) \leq p(x)$ for all x.

Let C be a class of functions. C is *closed under bounded general recursion* if whenever g, h, and p are in C then so is the function f as given above. (Bounded general recursion is actually bounded iteration (3.2.1), where all the functions involved are of one variable.)

We have from 3.2.11:

Proposition 4.1.4 Any function in $\mathcal{E}^n$ ($n \geq 3$) can be obtained from $s(x) = x+1$, $U_1(x,y) = x$, $U_2(x,y) = y$, $f_n^*(x,y)$, j, k, ℓ by repeatedly constructing a new function from previously obtained functions by substitutions and bounded iterations of the form:

$$g(x,0) = ax$$
$$g(x,sy) = bg(x,y)$$
$$g(x,y) = h(x,y) \quad .$$

Let i be the identity function $i(x) = x$. Suppose that the functions $j(i,o)$ and $j(k,s\ell)$ defined by $j(i,o)(x) = j(i(x),o(x))$ and $j(k,s\ell)(x) = j(k(x),s\ell(x))$ are in $\mathcal{E}^{n_j}$.

The next result shows that bounded iteration can be replaced by bounded general recursion of a special type.

Proposition 4.1.5 In the definition of $\mathcal{E}^n$ ($n \geq n_j$), bounded iteration can be replaced by bounded general recursion of the form:

$$fj(i,o) = a$$
$$fj(k,s\ell) = bf$$
$$f(x) \leq c(x) \quad .$$

Proof: Let g be defined by bounded iteration:

$$g(x,0) = ax$$
$$g(x,sy) = bg(x,y)$$
$$g(x,y) \leq h(x,y) \quad , \qquad a,b,h \in \mathcal{E}^n.$$

Then g can be defined by bounded general recursion as follows:

Define f:

$$fj(i,o) = a$$
$$fj(k,s\ell) = bf$$
$$f(x) \leq h(kx,\ell x)$$

Then $$g(x,y) = fj(x,y) .$$

Similarly, given a function f defined by bounded general recursion, we can define f using bounded iteration:

Suppose f is defined by:

$$fj(i,o) = a$$
$$fj(k,s\ell) = bf$$
$$f(x) \leq c(x) \quad , \qquad a,b,c \in \mathcal{E}^n .$$

Let

$$g(x,0) = a(x)$$
$$g(x,sy) = bg(x,y)$$
$$g(x,y) \leq cj(x,y)$$

Then, $$f(x) = g(kx,\ell x) .$$ □

The functions $f_n^*(x,y)$ were defined by iteration. But we could have defined functions $f_n'(x)$ by general recursion as in the proof of 4.1.5, so that $f_n^*(x,y) = f_n'j(x,y)$. Thus, the functions $f_n'(x)$ can replace $f_n^*(x,y)$ in the definition of $\mathcal{E}^n$ $(n \geq n_j)$. .

We now wish to consider functions of one variable in $\mathcal{E}^n$. This class will be denoted $\mathcal{E}_1^n$.

Proposition 4.1.6 (J. Robinson [1968]) Every function in $\mathcal{E}_1^n$ $(n \geq n_j)$ can be obtained from o, s, k, and ℓ by repeatedly constructing a new function f from previously obtained functions a and b by composition, pairing ($f = j(a,b)$), or general recursion of the form $fj(i,o) = a$; $fj(k,s\ell) = bf$.

Proof: First, the initial function $f_n'(x)$ can be so constructed. Hence (by 4.1.5) every function in $\mathcal{E}^n$ can be obtained from j, k, ℓ by general recursion of the form of the proposition, and substitution. To obtain functions of one variable by substitution, it is always sufficient to use substitution to define only functions of one variable. The only initial function of more thanone variable which yields new functions of one variable is j. Hence, besides compositions of functions of one variable, we must allow $f = j(a,b)$ to be constructed from previously obtained functions a and b. Hence all functions in $\mathcal{E}^n$ $(n \geq n_j)$ of one variable are obtained from o, i, s, k, and ℓ by compositions, pairings, and general recursion of the form given in the proposition. Since $i = j(k,\ell)$, we can omit i from the initial functions. □

Proposition 4.1.7 (J. Robinson [1968]). Every function in $\mathcal{E}_1^n$ $(n \geq n_j)$ can be obtained from o and s by repeated compositions and general recursions from previously defined functions in $\mathcal{E}_1^n$.

Proof: Let R be the class of functions obtainable from o and s by composition and general recursion. Then, i, k, and ℓ are in R:

$$io = o \qquad is = si$$

$$kj(i,o) = io \qquad kj(k,s\ell) = ik$$

$$\ell j(i,o) = o \qquad \ell j(k,s\ell) = s\ell$$

R is closed under pairing (J. Robinson). Furthermore, to define functions in $\mathcal{E}_1^n$, only functions in $\mathcal{E}_1^n$ are needed.

Hence, every function in $\mathcal{E}_1^n$ $(n \geq n_j)$ belongs to R, provided there is a pairing function j such that $j(i,o)$ and $j(k,s\ell)$ are in R. One such j is given by $j(x,y) = (2x + 1)2^y-1$. For this j, $j(x,o) = 2x$, $j(x,sy) = 2j(x,y) + 1$. Hence $j(i,o)$ is the double function d and $j(k,s\ell)$ is sd. d belongs to R since it is defined by the general recursion $do = o$, $ds = ssd$. Hence every function in $\mathcal{E}_1^n$ $(n \geq 3)$ belongs to R. □

It is possible that the theorem holds for $n \geq 2$, provided that there is a quadratic pairing function such that $j(i,o)$ and $j(k,s\ell)$ are easily definable.

A *functional equation* is an equation of the form $a_1 \dots a_k = b_1 \dots b_m$.

An equation $\alpha = \beta$ is *derivable* from a system E of functional equation if there is a chain of equations $\alpha_1 = \dots = \alpha_n$ $(n \geq 1)$ such that $\alpha_1 = \alpha$, $\alpha_n = \beta$, and for each t, α_{t+1} is obtained from α_t by replacing a part γ of α_t by δ, where $\gamma = \delta$ or $\delta = \gamma$ belongs to E.

A system $E(s,f,u_1,\dots)$ of functional equations *defines* a particular function f_0 if f_0 is the unique function f for which s is the successor function and there are functions $u_1, u_2, \dots$ so that $E(s,f,u_1,u_2,\dots)$ holds. Since every hyperarithmetical function can be defined by some system of functional equations (J. Robinson [1967]), there is in general no way to compute the values of a function defined by a system of functional equations.

Example 4.1.8 (J. Robinson [1968]) The equations

$$\begin{array}{ll} do = o & io = o \\ ds = ssd & is = si \end{array}$$

define the zero function $o(x) = 0$, the double function $d(x) = 2x$, and the identity function $i(x) = x$.

Proof: The equation ds = ssd gives: $d(x) = 2x + d(0)$. Thus, the only possible fixed point for d is 0. The equation do = o implies that Range(o) = 0. Hence o is identically 0 and $d(x) = 2x$. The equation is = si gives $i(x) = x + i(0)$. Since o is the zero function, we have that $i(0) = 0$. Therefore i is the identity function. □

Theorem 4.1.9 (J. Robinson [1968]) A function f of one variable is recursive if and only if there is a system $E(s,o,f,u_1,\dots)$ of functional equations such that

(i) E has a unique solution in which s is the successor function and o is the zero function. In this solution, f is the given function.

(ii) For all natural numbers n and m, the equation $fs^n0 = s^m0$ is derivable from E if and only if $f(n) = m$.

If (ii) holds, then we say that the values of f can be *derived* from E. To obtain E for a recursive function f, start with the equations $oo = o$ and $os = o$ and adjoin a sequence of equations corresponding to successive definitions of new functions by compositions and general recursions from o, s, and functions already defined, ending with a definition of f. This is possible by Theorem 4.1.2. E then satisfies (i) and (ii) so that given such an E we can effectively compute the values of f. It follows that equations of the form $fs^k o = s^m o$ are decidable; we simply systematically derive all possible equations from E. For each k, we will eventually derive an equation of the form $fs^k o = s^n o$. Thus, the equation $fs^k o = s^m o$ is derivable if and only if $m = n$. What happens to arbitrary expressions $a_1 \ldots a_k = b_1 \ldots b_m$ is not clear. It will be shown in the next section that equations true for the natural numbers are not derivable. We are interested in systems E of functional equations for the following reason:

The functions of E can be thought of as generators and the equations as relations of a semigroup, since a set of functions with the operation of composition is always a semigroup (i.e., an associative system). Derivability from a system of functional equations is analogous to derivability from the relations of a semigroup. Thus, our goal is to construct decidable semigroups by constructing decidable systems of functional equations.

4.2 Systems of Functional Equations--Decidability

Definition 4.2.1 Let E be a system of functional equations. An expression $f_1 \ldots f_k$ is called a *computation* if

(i) it is of the form $f_1 \ldots f_{k-1} o$, where o is the zero function and none of $f_1, \ldots, f_{k-1}$ are 'o';

or,

(ii) it can be transformed into an expression of the form $g_1 \ldots g_k o$ using the equations of E.

As stated in Section 4.1, if a system E of functional equations defines a recursive function f and E also defines all functions used in defining f, then all equations between computations are decidable (all such equations are equivalent to equations of the form $f_1 \ldots f_k s^m o = g_1 \ldots g_j s^n o$). We will now consider other equations, i.e., noncomputations. Here we give examples of very simple systems of functional equations from which equations true about the natural numbers are not derivable.

Example 4.2.2. Let E be the following system of equations:

(1) $oo = o$

(2) $os = o$

(3) $do = o$

(4) $ds = ssd$

These equations define both the zero function o and the double function d (4.1.8). We claim that od = o is not derivable. We show this by showing that o cannot be obtained from od using the above equations.

The only way 'd' can be removed is to use the equation 'do = o'. But there is no equation which can be used to get 'o' to the right of 'd'. Thus, 'od' cannot be transformed into 'o'.

□

However, note that $ods^k o = o$ is provable for all $k \geq 1$ ($ods^k o = os^{2k} o = o$), and $oso = oo = o$. Thus, here we have a situation where $f(x) = 0$ is provable for all natural numbers x, yet f = o (the zero function) is not provable.

In doing derivations from systems of functional equations E, we are dealing with formal systems. Indeed, the natural numbers N are a model of the equational system (as defined in Chapter 2) whose function symbols are those of E and whose axioms are the equations of E. Thus, the equation od = o is true if we consider d and o to be the double function and the zero function, respectively, defined on the natural numbers, but the fact that it is not provable means that there exist interpretations in which it is not true. The following is a possible interpretation for the system given in the example:

The domain consists of three distinct elements a_1, a_2, a_3. The functions o, d, s are defined as follows:

	o	d	s
a_1	a_1	a_1	a_1
a_2	a_2	a_3	a_2
a_3	a_3	a_3	a_3

The equations in the example are satisfied. However, $od(a_2) = o(a_3) = a_3$ and $o(a_2) = a_2$, i.e., $od(a_2) \neq o(a_2)$.

Following are some more examples. We show certain equations are not provable by constructing interpretations in which they are not true.

<u>Example 4.2.3</u>

$$\begin{array}{lll} oo = o & do = o & go = so \\ os = o & ds = ssd & gs = sg \end{array}$$

The functions defined are o, s, d, and $g(x) = x+1$. $g = s$ is not derivable. This is obvious because no relations are applicable, but it is something we might expect to be derivable, especially since $gs^k o = ss^k o$ is derivable for all k. The following is a model in which $g = s$ is not true.

	o	s	g	d
a_1	a_1	a_1	a_1	a_1
a_2	a_1	a_2	a_3	a_2
a_3	a_1	a_3	a_2	a_3

The equations above are satisfied, but $s(a_2) \neq g(a_2)$ and $s(a_3) \neq g(a_3)$.

Example 4.2.4

$$oo = o \qquad do = o \qquad io = o$$
$$os = o \qquad ds = ssd \qquad is = si$$

(i is the identity function). id = d is not derivable:

	i	o	d	s
a_1	a_1	a_1	a_1	a_1
a_2	a_3	a_1	a_2	a_1
a_3	a_1	a_1	a_3	a_3

Here $id(a_2) = i(a_2) = a_3$ and $d(a_2) = a_2$, i.e., $id(a_2) \neq d(a_2)$.

4.3 $\mathcal{E}^n$ -Decidable Semigroups

Let E be a system of functional equations defining some recursive function f and all the auxiliary functions needed to define f. For such a system, all equations of the form $gs^k o = s^m o$ (and therefore all computations) are decidable. In what follows, we show how to modify E so that it still computes the same functions, but so that *all* equations are decidable. The idea is to modify the defining equations of the given system so as to be able to distinguish between computations and noncomputations, and avoid indefinitely long derivations involving noncomputations, while not changing any equations $gs^i o = s^j o$ derivable from the original system, i.e., to change the system in such a way that N is still a model.

We remark that the method used in the previous section to show that certain equations are not derivable is not really practical: while all models constructed there were simple, in general, there is no a priori bound on the cardinality of the model we need, even more, there is no guarantee that the model is even finite.

Let E be a system of functional equations defining the recursive function f and all the functions needed to define f. Further, suppose E includes the following equations:

(1) $io = o$ ⎫ definition of the

(2) $is = si$ ⎭ identity function $i(x) = x$

(3) $of_n = o$ for all functions f_n in E

(4) $if_n = f_ni$ for all functions f_n in E

Let E' be the system with the same function symbols as E and the following equations: For all equations $\alpha = \beta$ in E ($\alpha = \beta$ is $a_1 \ldots a_m = b_1 \ldots b_n$)

(i) if $\alpha = \beta$ does not involve o and $n < m$, then $a_1 \ldots a_m = b_1 \ldots b_n i^{m-n} \in E'$.

(ii) If $\alpha = \beta$ does not involve o and $m < n$, then $a_1 \ldots a_m i^{n-m} = b_1 \ldots b_n \in E'$.

(iii) Otherwise, $\alpha = \beta \in E'$.

By the remarks at the end of Section 4.1, we can associate with E a semigroup. We will interchangeably refer to E as a system of functional equations or as a semigroup. Also, the terms "equations" and "relations" will be used interchangeably.

<u>Proposition 4.3.1</u> $E \vdash fs^io = s^jo \Leftrightarrow E' \vdash fs^io = s^jo$.

<u>Proof</u>: (⇒) Suppose $E \vdash fs^io = s^jo$.

The relations of E' differ from those of E in the number of i's present. Thus, to obtain a proof of $fs^io = s^jo$ from E', first use (1) to generate enough i's, then use the same proof as from E, with the corresponding relations in E', using (4) to move the i's into position and at the end, using (1) to remove excess i's.

(⇐) Suppose $E' \vdash fs^io = s^jo$.

To obtain a proof from E, use the relations in E corresponding to those in the proof from E', using (1), (4), and (3) to generate, move, and remove i's. □

Note that the only difference between a proof from E and a proof from E' of $fs^io = s^jo$ is in the number of i's present. N is a model of E' with i interpreted as the identity function. We remark that $if_n = f_n$, while true for N is probably' not provable.

Proposition 4.3.2 Let a_1 a_2 ... a_n be any expression on the function symbols of E'. Then it is decidable whether or not $a_1a_2...a_n$ is a computation.

Proof: By definition, $a_1...a_n$ is a computation if and only if it is equal to an expression of the form $b_1...b_mo$ under the relations of E'. We consider several cases:

(i) $a_1...a_n$ contains o.

Then either the spelling of $a_1...a_n$ is $a_1...a_{n-1}o$ (where none of $a_1...a_{n-1}$ are o) or $a_1...a_n$ is identically equal to $a_1...a_{k-1}oa_{k+1}...a_n$, where none of $a_1,...,a_{k-1}$ are o (some of $a_{k+1},...,a_n$ could be o). Then, by using relations (3), $a_1...a_n = a_1...a_{k-1}o$. Thus, $a_1...a_n$ is a computation.

(ii) $a_1...a_n$ does not contain o.

First note that if $\alpha = \beta$ is a relation in E' not involving o, then length(α) = length(β). Let k be the number of function symbols in E'. To dedide if $a_1...a_n$ is a computation, apply the relations of E' to $a_1...a_n$. Since the left-hand side and the right-hand side of relations not involving o have the same length, we have a bound on the maximum number of steps (i.e., the number of relations used) needed to reach an expression involving o: k^n (this is the number of distinct words of length n over an alphabet of k letters). If o does not appear within this number of steps, then $a_1...a_n$ is not a computation. □

Proposition 4.3.3 E' is decidable, i.e., there is an effective procedure to decide for any two words α, β whether $E' \vdash \alpha = \beta$.

Proof: The following is a procedure to decide whether $E' \vdash \alpha = \beta$: First check to see if both α and β are computations.

(i) If so, then both α and β are equal to expressions of the form $b_1 \ldots b_k o$. Since E' defines all the functions $b_1, \ldots, b_k$ and all the functions are recursive, the values of $b_1, \ldots, b_k$ are computable from E'. Thus, α can be transformed into an expression $s^{k_1}o$; β can be transformed into an expression $s^{k_2}o$, and so, $E' \vdash \alpha = \beta \Leftrightarrow k_1 = k_2$.

(ii) If not, then at least one of α and β is not a computation, say α. Then we can generate a list of everything derivable from α (this is finite and can be obtained in a finite number of steps), and $E' \vdash \alpha = \beta \Leftrightarrow \beta$ appears on this list.

It is clear that if $E' \vdash \alpha = \beta$, then this procedure will verify that, and if $E' \nvdash \alpha = \beta$, this will be shown in a finite number of steps. □

The difficulty of deciding whether $E' \vdash \alpha = \beta$ lies in the difficulty of computing the functions that E' defines. It is not difficult to decide whether $a_1 \ldots a_k$ is a computation. Furthermore, the fact that $\alpha = \beta$ is decidable when α and β are both computations depends on the fact that E' defines *recursive* functions. This suggests that if we had a bound on the computational complexity of the functions defined by E' we would have a bound on the degree of difficulty of E'.

Proposition 4.3.4 For $n \geq 4$, there exist strictly $\mathcal{E}^n$-decidable semigroups.

Proof: For $n \geq 4$, let S_n be a set in $\mathcal{E}^n - \mathcal{E}^{n-1}$ (cf. 3.1.9). By (3.1.6 (vi)), S_n and $\bar{S}_n$ (the complement of S_n) can be enumerated by functions f_n and $\bar{f}_n$ in $\mathcal{E}^0$. By (4.1.7) f_n and $\bar{f}_n$

can be defined from functions of one variable in $\mathcal{E}^3$. Let E_n be the system of functional equations defining f_n and $\bar{f}_n$ together with the equations:

$$gf_n = o$$
$$g\bar{f}_n = so\bar{f}_n$$

g is a 0-1 valued function defined by general recursion. Further, $gs^k o = o$ if and only if $k \in \text{range}(f_n)$, and $k \in \text{range}(f_n)$ if and only if $k \in S_n$, i.e., g is the characteristic function of S_n and therefore $g \in \mathcal{E}^n$. Thus, the word problem for E_n cannot be lower than $\mathcal{E}^n$, since a decision procedure for E_n could be used for computing S_n.

We now show that E_n is decidable at level $\mathcal{E}^n$. Suppose we wish to decide if $w_1 = w_2$. By 4.3.2 and 4.3.3, the number of steps (i.e., the number of relations that need to be used) to decide if w_1 and w_2 are computations and whether $w_1 = w_2$ when either w_1 or w_2 (or both) are not computations is an $\mathcal{E}^3$ function of $\max\{\text{length}(w_1), \text{length}(w_2)\}$.

Suppose w_1 and w_2 are both computations:

$$w_1 = g_1 \ldots g_k o$$
$$w_2 = h_1 \ldots h_m o$$

Let p(i) be the maximum number of steps needed to compute $fs^i o$ for any $f \in E_n$ $(p \in \mathcal{E}^n)$.

Let q(i) be the maximum number of steps needed to compute $fs^i o$ for any $f \in E_n$, $f \neq g$.

We can assume that p and q are strictly increasing. Since all the functions of E_n except g are in $\mathcal{E}^3$, $q \in \mathcal{E}^3$, and for all functions f in E_n ,

$$f(k) \leq q(k) \quad \text{for all } k$$

(easily seen to be true for all functions f not equal to g; it is true for g since g is 0-1 valued).

Thus, if f is an arbitrary function in $\mathcal{E}^n$, to compute $fs^i o$ requires $\leq p(i)$ steps and the length of the result is $\leq q(i)$. Using this fact k times, the maximum number of steps needed to compute $g_1 \ldots g_k o$ is

$$p(0) + pq(0) + pqq(0) + \cdots + pq^{k-1}(0) \leq kpq^{k-1}(0) \ .$$

Therefore, to decide "$w_1 = w_2$" when w_1 and w_2 are computations, we need $\leq kpq^{k-1}(0)$ steps, where $k = \max\{\text{length}(w_1), \text{length}(w_2)\}$, and this is in $\mathcal{E}^n$ $(n \geq 4)$ whenever p and q are.

□

For the purpose of constructing $\mathcal{E}^n$-decidable groups we need to modify the semigroups E_n. Let E_n^* be the following semigroup:

generators: $f_1,\ldots,f_k,q,q_1,h$

where $f_1,\ldots,f_k$ are the function symbols in E_n

relations:

(1) $q_1R_{i_1} = q_1R_{i_2}$ for all relations $R_{i_1} = R_{i_2}$ of E_n

(2) $q_1f_i = f_iq_1$ for $i = 1,2,\ldots,k$

(3) $hq_1oh = q$

(the h's are "end markers" and q_1 marks our place in the course of a derivation).

<u>Proposition 4.3.5</u> $E_n^* \vdash hq_1g_1\ldots g_moh = q \Leftrightarrow E_n \vdash g_1\ldots g_mo = o.$

<u>Proof</u>: (⇐) Suppose $E_n \vdash g_1\ldots g_mo = o$.
Then we can get a proof of $hq_1g_1\ldots g_moh = hq_1oh$ from E_n^* by using the relations of E_n^* corresponding to those of E_n used, and using (2) to move q_1 into position. Thus, $E_n^* \vdash hq_1g_1\ldots g_moh = hq_1oh$ and using (3), $E_n^* \vdash hq_1oh = q$.

(⇒) Suppose $E_n^* \vdash hq_1g_1\ldots g_moh = q$.
Then $E_n^* \vdash hq_1g_1\ldots g_moh = hq_1oh$. There exists a sequence $\alpha_1 \to \alpha_2 \to \ldots \to \alpha_j$ where α_1 is $hq_1g_1\ldots g_moh$ and α_j is hq_1oh. By removing h's and q_1 from the α's above and deleting duplicate words we have a proof from E_n. □

Proposition 4.3.6 If w is an arbitrary word in $E_n^\star$ and for all u in E_n, $u = o$ is $\mathcal{E}^n$-decidable (relative to E_n), then $w = q$ is $\mathcal{E}^n$-decidable.

Proof: Suppose w is not "q". Since h and q_1 are not introduced (except by (3)), if w is not of the form hAq_1Bh (A and B are words on f_i), then $w \neq q$. Therefore, suppose w is of this form. By 4.3.5, hAq_1Bh (i.e. w) $= q \Leftrightarrow AB = o$ in E_n, which is $\mathcal{E}^n$-decidable. □

CHAPTER 5

HNN EXTENSIONS

The Higman, Neumann, Neumann (HNN) extension, introduced by G. Higman, B. H. Neumann, and H. Neumann in 1949 is one of the basic constructions of group theory. It is used by Miller [1971] to prove various unsolvability results about groups, and a topological interpretation is given in Schupp [1973]. We are interested in HNN extensions because they provide a way of constructing a group G from a group G' in such a way that the word problem for G is reducible to the word problem for G'. In this chapter we examine the difficulty of the word problem for HNN extensions in general. In the next chapter, we will exhibit a sequence of groups G_n, each of which is a series of HNN extensions, and has word problem in $\mathcal{E}^n$.

The following notational conventions will be used: $(g_1,\ldots,g_n \mid R_1 = 1,\ldots, R_m = 1)$ denotes the group presented by generators $g_1,\ldots,g_n$, subject to the defining relations $R_1 = 1$, $R_2 = 1,\ldots, R_m = 1$. Generally, lower case letters will be used for generators of a group while upper case letters will be used for groups or for words in some specified group. If Q is the set of words in some group, <Q> will denote the subgroup generated by Q. If U and V are words of a group G, then $U \equiv V$ means that U and V are identical as words; $U = V$ means that U and V are equal in the group G.

5.1 Definitions and Main Results

Definition 5.1.1 (Rotman [1973]) Let $G = (S|D)$ be a group presentation. A second presentation G^* has *basis* G and *stable letters* $\{p_v \mid v \in V\}$ if

$$G^* = (S, p_v, \ v \in V \mid D, \ p_{v_j}^{-1} A_j p_{v_j} = B_j, \ j \in J)$$

where A_j and B_j are words on S.

For each $v \in V$ let $J(v)$ denote the following subset of the index set J:

$$J(v) = \{j \in J | p_v \text{ is involved in new relation } j\}$$

$$= \{j \in J \mid v = v_j \text{ and } p_v^{-1} A_j p_v = B_j\} \ .$$

Let A, B, A(v), and B(v) be the following subgroups of G:

$$A = \langle A_j \mid j \in J\rangle$$

$$B = \langle B_j \mid j \in J\rangle$$

$$A(v) = \langle A_j \mid j \in J(v)\rangle$$

$$B(v) = \langle B_j \mid j \in J(v)\rangle$$

Definition 5.1.2 (Rotman [1973]) The *isomorphism condition* is that for each $v \in V$, there is an isomorphism $\phi_v : A(v) \to B(v)$ under which $\phi_v(A_j) = B_j$ for all $j \in J(v)$.

Definition 5.1.3 (Rotman [1973]) A group G^* is an *HNN extension* of G with stable letters $\{p_v \mid v \in V\}$ if

(i) G^* has basis G and stable letters $\{p_v | v \in V\}$ (so presentations of G^* and G are given), and

(ii) the isomorphism condition holds.

Thus, an HNN extension is formed by adding inner automorphisms. Another definition, given for one stable letter t is:

Definition 5.1.4 Let G be a group, and A and B be subgroups of G with ϕ: $A \to B$ an isomorphism. The HNN extension of G relative to A, B, and ϕ is the group

$$G^* = \langle G, t \mid t^{-1} a\, t = \phi(a),\ a \in A \rangle$$

The importance of HNN extensions lies in the following:

Theorem 5.1.5 (Britton's Lemma) Let G^* be a Britton extension of $G = (S \mid D)$ with stable letters $\{p_v \mid v \in V\}$. If w is a word involving at least one stable letter and if $w = 1$ in G^*, then w contains a subword of the form $p_v^{-e} C p_v^{e}$ where $e = \pm 1$ and C is a word on S.

Moreover, if $e = +1$, then C is equal in G to a word on $\{A_j \mid j \in J(v)\}$ (i.e., C is equal in G to an element of $A(v) \subset A$); if $e = -1$, then C is equal in G to a word on $\{B_j \mid j \in J(v)\}$ (i.e., C is equal in G to an element of $B(v) \subset B$).

The combinatorial significance of stating "C is equal in G ..." is that C can be transformed into a word in A(v) or B(v) without inserting $p_v p_v^{-1}$ or $p_v^{-1} p_v$.

Definition 5.1.6 (Rotman [1973]) A word of the form $p_v^{e} C p_v^{-e}$ is called a *pinch* if $e = \pm 1$ and the "moreover" paragraph above holds for C.

Remark. Instead of "$w = 1$ in G^* " in the above theorem we could have "$w = T$ in G^*," where T is any word in G and the theorem would still hold.

Definition 5.1.7 (Rotman [1973]) Given G^* with basis G and stable letters $P = \{p_v \mid v \in V\}$ then a word w is p_v-*reduced* (for some fixed p_v) if w contains no pinch $p_v^{e} C p_v^{-e}$ as a subword. We say that w is p-*reduced* if w is p_v-reduced for all $p_v \in P$, p[w] will denote the p-reduced w.

It is easy to see that an arbitrary word w is equal in G^* to a p-reduced word: the relations of G^* allow us to replace each pinch by a word involving two fewer occurrences of a stable letter. Thus, if $w = 1$, then w can be p-reduced successively to obtain a word on S which equals 1. If w cannot be p-reduced, then $w \neq 1$ in G^*. This is a procedure (not necessarily effective) to solve the word problem for G^*. To state this more formally, we introduce the following notation:

$(?\ \beta)\ T(\beta)$ denotes the problem of deciding for an arbitrary word β, whether $T(\beta)$ is true

Let $\bar{A}(v)$ be a variable ranging over words in $A(v)$.

Let $\bar{B}(v)$ be a variable ranging over words in $B(v)$.

Proposition 5.1.8 The problem of obtaining $p[w]$ for $w \in G^*$ is reducible to

(i) $(?u \in G)(\exists \bar{A}(v))(u = \bar{A}(u))$ in G

(ii) $(?u \in G)(\exists \bar{B}(v))(u = \bar{B}(v))$ in G .

Proof: By definition of p-reduction. □

Proposition 5.1.9 Let T be a variable ranging over some particular set of words of G. Then $(?w \in G^*)(\exists T) w = T$ in G^* is reducible to

(1) $(?u \in G)\ \exists \bar{A}(v) u = \bar{A}(v)$ in G

$(?u \in G)\ \exists \bar{B}(v) u = \bar{B}(v)$ in G

(2) $(?u \in G)\ (\exists T) u = T$ in G .

Proof: Since any word is equal in G^* to a p-reduced word, we have $(\exists T)(w = T \text{ in } G^*) \Leftrightarrow (\exists T)(p[w] = T \text{ in } G^*)$.

We will now show that

$$(\exists T)(p[w] = T \text{ in } G^*) \Leftrightarrow p[w] \text{ is p-free and } (\exists T)(p[w] = T \text{ in } G) .$$

The implication $(\Leftarrow)$ is clear. To see $(\Rightarrow)$, suppose $p[w]$ is not p-free, but $(\exists T)(p[w] = T \text{ in } G^*)$. Then,

$p[w]T^{-1} = 1$ in G^* and Britton's lemma applies. Thus, $p[w]T^{-1}$ contains a subword $p_v^e C p_v^{-e}$ and C is of the required form. But $p_v^e C p_v^{-e}$ must be entirely contained in p[w] since T is p-free, contradicting the fact that p[w] is p-reduced. Therefore,

$$(\exists T)(p[w] = T \text{ in } G^*) \Rightarrow p[w] \text{ is p-free and } (\exists T)(p[w] = T \text{ in } G) \ . \qquad \square$$

Corollary 5.1.10 $w = 1$ in G^* is reducible to

(1) $(?u \in G)(\exists \bar{A}(v))(u = \bar{A}(v) \text{ in } G)$
$(?u \in G)(\exists \bar{B}(v))(u = \bar{B}(v) \text{ in } G)$

(2) $(?w \in G) w = 1 \text{ in } G$.

Proof: Take 1 as the range of T in 5.1.9. □

Proposition 5.1.11 (Rotman [1973]) Let G^* be a Britton extension of G with stable letters $P = \{p_v | v \in V\}$. Assume $U \equiv R_0 p_v^{e_1} R_1 \cdots p_v^{e_m} R_m$ and $W \equiv L_0 p_v^{f_1} L_1 \cdots p_v^{f_n} L_n$ are P-reduced words, where each e, f = $\pm$ 1 and none of the (possibly empty) words on R or L involve p_v. If $U = W$ in G^*, then $m = n$ and $(e_1, \ldots, e_m) = (f_1, \ldots, f_n)$. Moreover, the word $p_v^{e_m} R_m L_n^{-1} p_v^{-f_n}$ is a pinch.

5.2 $\mathcal{E}^n$-Decidability of HNN Extensions

We will now discuss the $\mathcal{E}^n$ solvability of the word problem for G^*. Suppose (for the rest of this chapter) that G^* is finitely presented. Then there are only a finite number of relations $p_{v_j}^{-1} A_j p_{v_j} = B_j$ so that if we have a word of the form $p_{v_j}^{-1} A^* p_{v_j}$ where $A^* \equiv$ word on $A(v_j)$ and we p_{v_j}-reduce, then there is a constant c such that length(B^*) (where $p_{v_j}^{-1} A^* p_{v_j} = B^*$) is $\leq c \cdot$length(A^*).

Proposition 5.2.1 Let

$$\left.\begin{array}{l}(?U \in G)(\exists \bar{A}(v))(U = \bar{A}(v) \text{ in } G)\\ \\ (?U \in G)(\exists \bar{B}(v))(U = \bar{B}(v) \text{ in } G)\end{array}\right\} \qquad (*)$$

be decidable at level $\mathcal{E}^n$. Then the number of steps needed to obtain p[w] is in $\mathcal{E}^{n+1}$.

Proof: Assume that if $U = \bar{A}(v)$ in G, then we can explicitly obtain a representation U_A of U in terms of the elements of A(v) in $\leq$ f(length(U)) steps, $f \in \mathcal{E}^n$. Assume also that f is an increasing function.

In order to p-reduce once, $\leq$ f(length(U)) steps are required, the length of U_A is $\leq$ f(length(U)) and by the above remark, the length of the new word is $\leq$ cf(length(U)). p-reducing takes the most number of steps when U is of the form $p_{v_1}^{-1}U_1p_{v_2}^{-1}U_2p_{v_3}^{-1}\ldots p_{v_3}U_2'p_{v_2}U_1'p_{v_1}$, for then the length of the word to which the procedure for (*) is applied depends on the result of the previous application, i.e., to p-reduce U twice can require

$$\leq \underbrace{f(\text{length}(U))}_{\substack{\text{to reduce}\\ \text{once}}} + \underbrace{f(cf(\text{length}(U)))}_{\substack{\text{to reduce a}\\ \text{second time}}} \leq 2f(cf(\text{length}(U))) \text{ steps}$$

and the length of the result is $\leq$ cf(cf(length(U))). Thus, to p-reduce m times, we need

$$\leq m(cf)^m(\text{length}(U)) = g(m, \text{length}(U)) \text{ steps}$$
$$\text{and } g \in \mathcal{E}^{n+1} \qquad (3.2.2)$$

The maximum number of times a word U can be p-reduced is $\leq$ length(U). Thus, to p-reduce U requires $\leq$ g(length(U), length(U)) steps, which is in $\mathcal{E}^{n+1}$. □

There is an analogy between p-reducing words and functions defined by iteration from functions in $\&^n$.

Let h(x,y) be the following function:

$$h(x,0) = f(x)$$

$$h(x,y+1) = gh(x,y)$$

i.e.,

$$h(x,y) = g^y(f(x)) .$$

If f and g are in $\&^n$, then h is in $\&^{n+1}$ (cf. 3.2.2). The complexity of h here depends on the size of g. If g is nondecreasing, then g is being applied to larger and larger arguments, causing h to grow very fast.

As seen in the proof of 5.2.1, the procedure for p-reducing might have to be applied iteratively to larger and larger words, each resulting from a previous application of the procedure. This causes the decision procedure to be in $\&^{n+1}$.

Suppose in the above definition of h we have a bound on g, $g(x) \leq e(x)$, $e \in \&^{n-1}$, e nondecreasing. Then

$$h(x,y) = g^y f(x) \leq g^{y-1} ef(x) \leq e^y f(x) \in \&^n \qquad (3.2.2)$$

i.e., h(x,y) is bounded by a function in $\&^n$. Therefore, $h(x,y) \in \&^n$.

Similarly, we will show that if after p-reducing a word w once, the length of the resulting word is $\leq$ f(length(w)) and $f \in \&^{n-1}$, then the procedure for p-reducing is in $\&^n$.

<u>Corollary 5.2.2</u> If the length of U_A in 5.2.1 is $\leq$ h(length(U)) where h is a function in $\&^{n-1}$, then the number of steps needed to obtain p[w] is in $\&^n$.

<u>Proof</u>: Using a proof similar to the one above, we can see that the number of steps needed to p-reduce w m times is $\leq mfh^{m-1}$(length(w)) which is in $\&^n$ (3.2.2). □

Corollary 5.2.3 If the relations involving the stable letters p_v of G^* are of the form $p_{v_j}^{-1}Ap_{v_j} = A$, then the number of steps needed to obtain p[w] is in $\mathcal{E}^n$.

Proof: If $w = U'p_{v_j}^{-1}U''p_{v_j}U'''$ and $U'' \in A(v)$, then we can apply the procedure for (*) in 5.2.1 to the word $U'U''U'''$ (i.e., the length of the word does not increase, h(x) = x in 5.2.2. □

Theorem 5.2.4 If G has word problem in $\mathcal{E}^n$ and the assumptions of (5.2.2) or (5.2.3) hold, then G^* has word problem in $\mathcal{E}^n$.

Proof: Suppose that to p-reduce $U \in G^*$ requires $\leq$ f(length(U)) steps, $f \in \mathcal{E}^n$, nand to decide if w = 1 in G requires $\leq$ g(length(w)) steps, $g \in \mathcal{E}^n$. Then to solve the word problem for G^* requires, for $U \in G^*$ $\leq$ f(length(U))+gf(length(U)) steps (by 5.1.10). □

Remark 5.2.5 In order to p-reduce a word in G^*, we have to solve a "subgroup problem". This may be very difficult. We show in the next chapter that there exist groups G and G* such that G has easily solvable word problem, but G* has more difficult or even unsolvable word problem (i.e., G has a more difficult or unsolvable subgroup problem).

Proposition 5.2.6 Let $G = (g, t \mid t^{-1}g\,t = g^{-1})$. Then G has word problem in $\mathcal{E}^2$.

Proof: G is an HNN extension of a free group on one generator. But G is also the semidirect product of two free groups. By 3.3.5 G has word problem in $\mathcal{E}^2$. □

Proposition 5.2.7 Let $G = (g,t \mid t^{-1}gt = g^2)$. Then G has word problem in $\mathcal{E}^3$.

Proof: G is an HNN extension of a free group with stable letter t.

Let w be an arbitrary word in G. To t-reduce w, we need to decide the following two questions (by 5.1.8):

(i) $(?u \in \langle g \rangle)(\exists v \in \langle g \rangle) u = v$ in $\langle g \rangle$

(ii) $(?u \in \langle g \rangle)(\exists v \in \langle g^2 \rangle) u = v$ in $\langle g \rangle$

(i) is trivial; it is always true; and we can obtain a freely reduced representation of u by scanning from left to right and reducing $gg^{-1} = 1$ and $g^{-1}g = 1$. The number of relations used is $\leq$ length(u). By 3.3.2, the procedure is in $\mathcal{E}^2$. To decide (ii), the procedure is similar to (i); just scan the word from left to right, replacing gg^{-1} and $g^{-1}g$ by 1. The number of relations used is $\leq$ length(U). Then just count and check whether or not the number of g's or g^{-1}'s in the remaining word is a multiple of two.

Thus, the procedure for (i) and (ii) is in $\mathcal{E}^2$. By 5.2.1, the procedure to t-reduce w is in $\mathcal{E}^3$. By the proof of 5.2.4, and by 3.3.3, G has word problem in $\mathcal{E}^3$. □

Remark 5.2.8 By the same proof as above, we can show that the following group has word problem in $\mathcal{E}^3$:

$$G = (g, t_1, \ldots, t_n \mid t_i^{-1} g t_i = g^2 \text{ for } i = 1, \ldots, n)$$

CHAPTER 6

$\mathcal{E}^n$-DECIDABLE GROUPS

The word problem was proved to be recursively unsolvable by constructing a group which modeled derivations in a semigroup obtained from a Turing machine computing a nonrecursive function. We use the same scheme for obtaining a presentation of a group, given the presentation of a semigroup, and we show that if the semigroup is such that $w = q$ (w is an arbitrary word on the generators of the semigroup, q is a fixed generator) is $\mathcal{E}^n - \mathcal{E}^{n-1}$ decidable ($n \geq 4$), then the group is $\mathcal{E}^n - \mathcal{E}^{n-1}$ decidable. The techniques used are similar to those in Boone [1964]. The group is a series of HNN extensions $G_0 \subseteq G_1 \subseteq G_2 \subseteq G_3 \subseteq G_4 = G$; we show that problems in G_i are reducible to problems in G_{i-1}, and furthermore, if the problems in G_{i-1} are in $\mathcal{E}^n$, then the problems for G_i are in $\mathcal{E}^n$. The construction of the groups and an indication of the proof of $\mathcal{E}^n$-decidability are given in 6.1, Section 6.2 contains the proofs, and 6.3 contains some conclusions.

6.1 Construction of $\mathcal{E}^n$-Decidable Groups ($n \geq 4$)

For $n \geq 4$, let S' be the semigroup given at the end of Section 4.3. By relabeling the generators of S so that they are $q, q_1, f_1, \ldots, f_M$ (i.e., reindexing the f's and letting h be some f_k), we have a semigroup S with presentation:

$$S = (q, q_1, f_1, \ldots, f_M \mid F_i q_{i_1} G_i = H_i q_{i_2} K_i,\ i \in I)$$

where F_i, G_i, H_i, K_i are (possibly empty) words on $C = \{f_b \mid b = 1, \ldots, M\}$, $q_{i_1}, q_{i_2} \in \{q, q_1\}$ and S is such that for an arbitrary word w on the generators of S, $w = q$ is $\mathcal{E}^n$-decidable.

Remark 6.1.1 We note that by using the method of constructing a new semigroup S_2 from a given semigroup S_1 given at the end of Section 4.3, any semigroup can be made to fit into this scheme (o in relation (3) would be replaced by some word u on the generators of S_1). We would then have $w = u$ in S_1 $\Leftrightarrow$ $hq_1wh = q$ in S_2.

We now present the scheme for constructing groups which will be seen to have word problem in $\mathcal{E}^n$ ($n \geq 4$). For each choice of a semigroup S as above, G is the following group:

generators: $q, q_1, f_1, \ldots, f_M, r_i$ $(i \in I)$, x, t, k

relations: (for all $i \in I$ and $b = 1, \ldots, M$)

$$\left.\begin{array}{l}
\left.\begin{array}{l}
\left.\begin{array}{l}
\left.\begin{array}{l} f_b^{-1} x f_b = x^2 \end{array}\right] \Delta_1 \\
r_i^{-1} f_b x r_i = f_b x^{-1} \\
r_i^{-1} \bar{F}_i q_{i_1} G_i r_i = \bar{H}_i q_{i_2} K_i
\end{array}\right] \Delta_2 \\
t^{-1} r_i t = r_i, \quad t^{-1} x t = x
\end{array}\right] \Delta_3 \\
k^{-1} r_i k = r_i, \quad k^{-1} x k = x \\
k^{-1} q^{-1} t q k = q^{-1} t q
\end{array}\right] \Delta_4$$

The following notation will be used: If $X = f_1^{e_1} \ldots f_m^{e_m}$ where $e_i = \pm 1$ (i.e., X is a word on C), then $\bar{X} = f_1^{-e_1} \ldots f_m^{-e_m}$. If X and Y are words on C, then $(Xq_jY)^* = \bar{X}q_jY$. Σ is *special* if $\Sigma \equiv \bar{X}q_jY$ where X and Y are positive words (i.e., no negative exponents) on C and $q_j \in \{q, q_1\}$.

Lemma 6.1.2 (Boone) If Σ is a special word, then $k\Sigma^{-1}t\Sigma = \Sigma^{-1} t \Sigma k$ in G if and only if $\Sigma^* = q$ in S.

Corollary 6.1.3 G cannot have word problem solvable at level $< \mathcal{E}^n$.

Proof: If the word problem for G were solvable at level $< \mathcal{E}^n$, then we would have a procedure in level $< \mathcal{E}^n$ to determine for an arbitrary special word Σ, whether $\Sigma^* = q$ in S. By 4.3.4 and 4.3.6 this cannot happen. □

Define the following groups:

$$G_0 = \langle x \rangle, \quad \text{the free group on } x$$
$$G_1 = (G_0, f_b,\ b = 1, \ldots, M \mid \text{relations } \Delta_1)$$
$$G_2 = (G_1, r_i,\ i \in I,\ q, q_1 \mid \text{relations } \Delta_2)$$
$$G_3 = (G_2,\ t \mid \text{relations } \Delta_3)$$
$$G = (G_3, k \mid \text{relations } \Delta_4)$$

Theorem 6.1.4 (Rotman [1973])

(i) G_1 is an HNN extension of G_0 with stable letters

$$\{f_b,\ b = 1, \ldots, M\}$$

(ii) $G_1 * \langle q, q_1 \rangle$ is an HNN extension of G_1 with stable letters

$$\{q, q_1\}$$

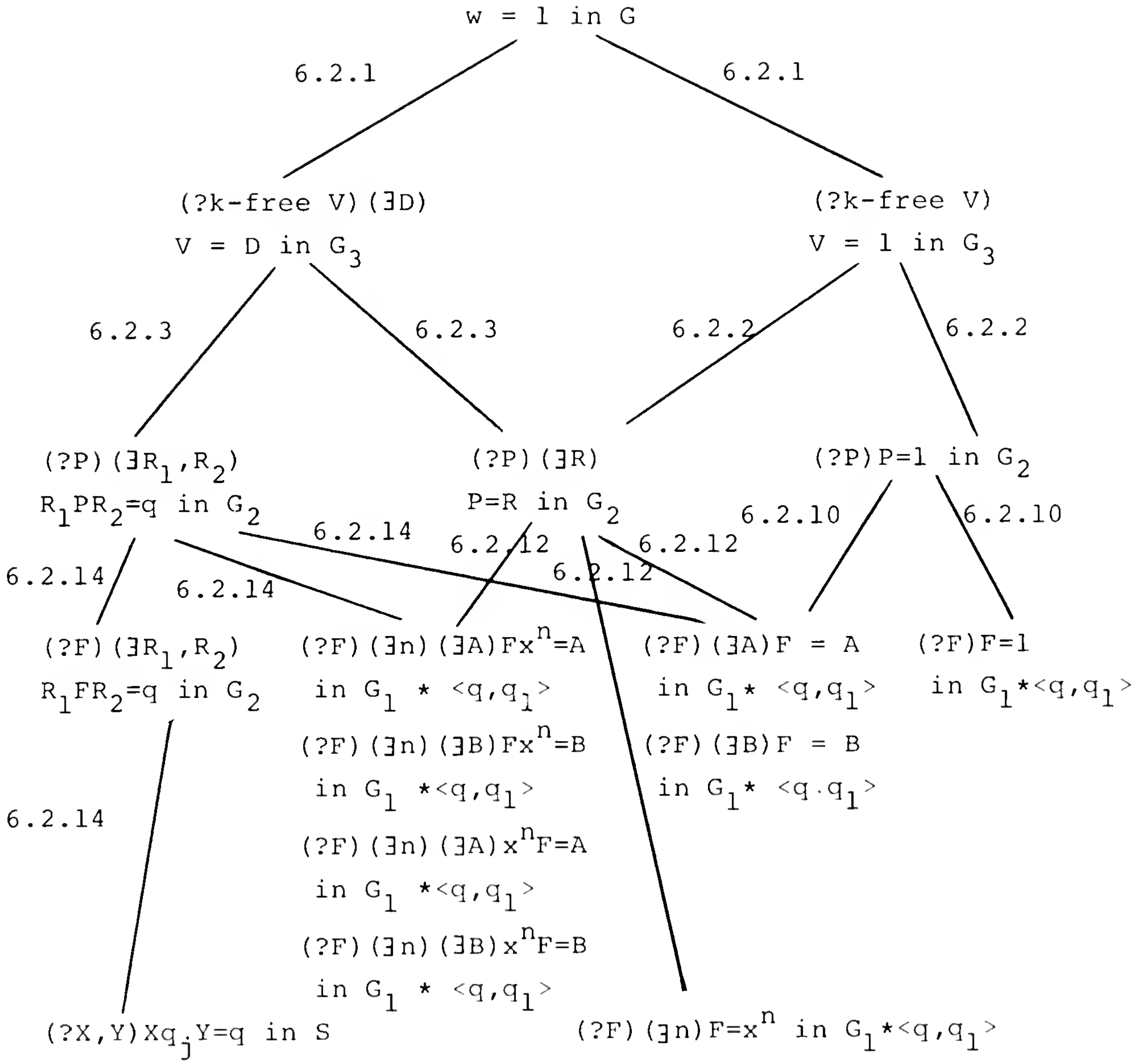
(iii) G_2 is an HNN extension of $G_1 * \langle q, q_1 \rangle$ with stable letters

$$\{r_i,\ i \in I\}$$

(iv) G_3 is an HNN extension of G_2 with stable letter t

(v) G is an HNN extension of G_3 with stable letter k.

The proof that G has word problem in $\mathcal{E}^n$ will consist of several reductions. These are shown schematically on the following page. The proofs will generally consist of two parts. We will show that a particular problem is reducible to simpler problems, and also, that if the simpler problems are $\mathcal{E}^n$-decidable, then so is the original problem. Ultimately, a "simple problem" will be shown to be $\mathcal{E}^n$-decidable, thus making all the problems $\mathcal{E}^n$-decidable. We recall that the notation $(?\beta)T(\beta)$ denotes the problem to determine for an arbitrary word β whether $T(\beta)$ is true; if G^* is an HNN extension of G with stable letters p_v and w is any word in G^*, then $p[w]$ is the p-reduced word w; $\|w\|$ is the word w with aa^{-1} and $a^{-1}a$ deleted. $U \equiv V$ means that U and V have exactly the same spelling.


$w = 1$ in G
6.2.1
6.2.1
$(?k\text{-free } V)(\exists D)$ $V = D$ in G_3
$(?k\text{-free } V)$ $V = 1$ in G_3
6.2.3
6.2.3
6.2.2
6.2.2
$(?P)(\exists R_1, R_2)$ $R_1PR_2 = q$ in G_2
$(?P)(\exists R)$ $P = R$ in G_2
$(?P)P=1$ in G_2
6.2.14
6.2.14
6.2.14
6.2.12
6.2.12
6.2.12
6.2.10
6.2.10
$(?F)(\exists R_1, R_2)$ $R_1FR_2 = q$ in G_2
$(?F)(\exists n)(\exists A)Fx^n=A$ in $G_1 * <q,q_1>$
$(?F)(\exists n)(\exists B)Fx^n=B$ in $G_1 * <q,q_1>$
$(?F)(\exists n)(\exists A)x^nF=A$ in $G_1 * <q,q_1>$
$(?F)(\exists n)(\exists B)x^nF=B$ in $G_1 * <q,q_1>$
$(?F)(\exists A)F = A$ in $G_1 * <q,q_1>$
$(?F)(\exists B)F = B$ in $G_1 * <q \cdot q_1>$
$(?F)F=1$ in $G_1 * <q,q_1>$
6.2.14
$(?X,Y)Xq_jY=q$ in S
$(?F)(\exists n)F=x^n$ in $G_1 * <q,q_1>$

We designate certain variables to range over special types of words:

Variables	Range
D	products of the words $x, r_i, q^{-1}tq$ (and inverses)
P	k,t-free words of G, i.e., words of G_2
R, R_1, R_2	words on x and r_i , $i \in I$
F	k,t,r_i $(i \in I)$ -free words of G, i.e., words of $G_1 * \langle q, q_1 \rangle$
A	products of the words $\bar{F}_i q_{i_1} Gi$, $f_b x$, $b = 1,\ldots,M$ (i fixed)
B	products of the words $\bar{H}_i q_{i_2} K_i$, $f_b x^{-1}$ $b = 1,\ldots,M$ (i fixed)
M, N	k,t,q,r_i-free words of G, i.e. words of G_1
X, Y	positive words on f_b , $b = 1,\ldots,M$

V and w will stand for arbitrary words of G.

6.2 Proofs

For the rest of this section, we take $n \geq 3$.

Lemma 6.2.1 (i) $w = 1$ in G is reducible to

(1) $(?k\text{-free } V)(\exists D)V = D$ in G_3

(2) $(?k\text{-free } V)V = 1$ in G_3

(ii) If (1) and (2) are $\mathcal{E}^n$-decidable, then G has word problem in $\mathcal{E}^n$.

Proof: (i) 5.1.10

(ii) 5.2.4 (and 5.2.3). □

Lemma 6.2.2 (i) $(?k\text{-free } V)V = 1$ in G_3 is reducible to

(1) $(?P)(\exists R)P = R$ in G_2

(2) $(?P)P = 1$ in G_2

(ii) If (1) and (2) are $\mathcal{E}^n$-decidable, then G_3 has word problem in $\mathcal{E}^n$.

Proof: (i) 5.1.10.

(ii) 5.2.4 (and 5.2.3). □

Lemma 6.2.3 (i) $(?k\text{-free } V)(\exists D)(V = D$ in $G_3)$ (*) is reducible to

(1) $(?P)(\exists R)P = R$ in G_2

(2) $(?P)(\exists R_1, R_2)R_1 P R_2 = q$ in G_2

(ii) If (1) and (2) are $\mathcal{E}^n$-decidable, then (*) is $\mathcal{E}^n$-decidable.

Proof: (i) Given an arbitrary k-free word V, first t-reduce V (this is possible using (1)). Let $t[V]$ be the t-reduced word. Then, $(\exists D)V = D$ in $G_3 \Leftrightarrow (\exists D)t[V] = D$ in G_3. Since every word in G_3 is equal to some t-reduced word, we can take D to be t-reduced. There are two cases to consider:

(a) $t[V]$ is t-free.
Then $t[V] = D$ and D t-reduced $\Rightarrow D = R =$ word on r_i and x.
Thus, if $t[V]$ is t-free, then

$$(\exists D)t[V] = D \text{ in } G_3 \Leftrightarrow (\exists R)t[V] = R \text{ in } G_2$$

(b) $t[V]$ is not t-free.
Then $t[V] = Ut^eP$ where $e = \pm 1$, P is t-free.

(**) If $Ut^eP = D$ (D t-reduced), then by 5.1.11,
$D \equiv D'q^{-1}t^eqR_1$ and $PR_1^{-1}q^{-1}$ is a pinch; i.e.,
$PR_1^{-1}q^{-1} = R$ in G_2, where R is a word on r_i and x.
Hence, $R^{-1}PR_1^{-1} = q$.

(***) On the other hand, if $(\exists R_1,R_2)R_1PR_2 = q$, then

$$Ut^eP = UPP^{-1}t^eP = UPR_2(R_2^{-1}P^{-1}R_1^{-1})R_1t^ePR_2R_2^{-1}$$
$$= UPR_2q^{-1}t^eqR_2^{-1}.$$

And so, $(\exists D)Ut^eP = D \Rightarrow (\exists D)UP = D$.
But also, $(\exists D)UP = D \Rightarrow (\exists D)Ut^eP = D$.
Hence, $(\exists D)Ut^eP = D \Leftrightarrow (\exists D)UP = D$.

Thus, for $t[V]$ not t-free, the procedure to decide (*) would be as follows: Decide (using (2)) for $t[V] = Ut^eP$ if $R_1PR_2 = q$ in G_2 for some R_1 and R_2. If not, then by (**), *not* $(\exists D)V = D$ in G_3. Otherwise, apply the same procedure to UP (by (***), the number of t's in the word to which the procedure is applied decreases by one with each application of the procedure). Do this until there are no more t's and then use (1).

(ii) By (5.2.3), if (1) is $\mathcal{E}^n$-decidable, then V can be t-reduced in $\leq f(\text{length}(V))$ steps, $f \in \mathcal{E}^n$. Note that since the relations involving t are $txt^{-1} = x$, $tr_it^{-1} = r_i$, we can t-reduce merely by deleting t's and $\text{length}(t[V]) \leq \text{length}(V)$.

The number of times (2) needs to be decided depends on the number of t's. Also, by (**), the word for which (2) needs to be decided is a subword of the original word with t's removed. Therefore, if (2) is decidable in time
$\leq$ g(length(V)), then (*) is decidable in time
$\leq$ f(length(V)) + (number t's)(g(length(V)))
$\leq$ f(length(V)) + (length(V))(g(length(V))), which is in $\mathcal{E}^n$
if f and g are. □

Lemma 6.2.4 Let M be a word on f_b and x, i.e., $M \in G_1$. Let $C = \{f_b | b = 1,\ldots,M\}$. Then the number of steps needed to obtain f[M] (the f_b-reduced word for all b) is in $\mathcal{E}^3$. □

Proof: (5.2.8).

Lemma 6.2.5 (i) Let A be a word on $\{\bar{F}_i q_{i_1} G_i, f_1 x, \ldots, f_M x\}$ (i fixed). Then $A' \equiv \Vert A \Vert$ is f-reduced and q-reduced.

(ii) Let B be a word on $\{\bar{H}_i q_{i_2} K_i, f_1 x^{-1}, \ldots, f_M x^{-1}\}$ (i fixed). Then $B' \equiv \Vert B \Vert$ is f-reduced and q-reduced.

Proof: Inspection of the spelling of A' and B' shows that A' and B' can contain no subword $f_k x^{2n} f_k^{-1}$ or $f_k^{-1} x^n f_k$. □

Remark 6.2.6 By 5.1.11, if F is f- and q-reduced and $(\exists n) x^n F = A$ or $(\exists n) F x^n = A$ (where A is f- and q-reduced), then the sequence of q_i's and f_b's in F and A must be identical.

From 6.2.5, we can conclude the following:

Lemma 6.2.7 Let w^* be the word w with x's deleted. If F is f- and q-reduced and F contains q_i^e ($e = \pm 1$), but no subword F' where $(F')^* = (\bar{F}_i q_{i_1} G_i)^e$ then

(i) $(\nexists n)\ x^n F = A$ in $G_1 * \langle q, q_1 \rangle$

(ii) $(\nexists n)\ F x^n = A$ in $G_1 * \langle q, q_1 \rangle$

(iii) $(\nexists A)\ F = A$ in $G_1 * \langle q, q_1 \rangle$.

(An analogous statement holds with "A" replaced by "B" and "$\bar{F}_i q_{i_1} G_i$" replaced by "$\bar{H}_i q_{i_2} K_i$").

Note: If F is f-reduced, then $\|F\|$ is q-reduced.

Lemma 6.2.8 Each of the following is $\mathcal{E}^3$-decidable:

(i) $(?F)(\exists n)(\exists A)\ Fx^n = A$ in $G_1 * \langle q,q_1\rangle$

(ii) $(?F)(\exists n)(\exists B)\ Fx^n = B$ in $G_1 * \langle q,q_1\rangle$

(iii) $(?F)(\exists n)(\exists A)x^nF = A$ in $G_1 * \langle q,q_1\rangle$

(iv) $(?F)(\exists n)(\exists B)\ x^nF = B$ in $G_1 * \langle q,q_1\rangle$.

Proof: A proof will be given for (i). All the rest are similar. To decide (i), first f- and q-reduce F. Let $[F]$ be the f- and q-reduced word. Check if $[F]$ contains a subword of the form given in 6.2.7 whenever $[f]$ contains q_{i_1}. If not, then *not* $(\exists n)(\exists A)Fx^n = A$. If so (or if F contains no q_{i_1}), then by 6.2.6 we know that word A Fx^n has to be equal to if indeed $(\exists n)(\exists A)Fx^n = A$. Thus, we can generate a word A_1 whose sequence of q's and f's is the same as that of $[F]$ and check if $A_1^{-1}F = x^n$.

To show $\mathcal{E}^3$-decidability, suppose that to f- and q-reduce F requires $\leq g(\text{length}(F))$ steps. If $(\exists n)(\exists A)\ Fx^n = A$, then $\text{length}(A) \leq 2\ \text{length}(F)$, for the number of f's in A is $\leq$ the number of f's in F, and the number of x's in A is $\leq$ the number of f's in A. Thus, to decide "$A_1^{-1}[F] = x^n$" requires $\leq g(g(\text{length}(F)) + 2\cdot\text{length}(F))$ steps, and so to decide (i) requires $\leq 2\cdot g(2\cdot g(\text{length}(F)))$ steps, which is in $\mathcal{E}^3$. □

Corollary 6.2.9 The following are $\mathcal{E}^3$-decidable:

(i) $(?F)(\exists A)F = A$ in $G_1 * \langle q,q_1\rangle$

(ii) $(?F)(\exists B)F = B$ in $G_1 * \langle q,q_1\rangle$.

Proof: Same as 6.2.8, except that instead of checking $A_1^{-1}F = x^n$, we check whether $A_1^{-1}F = 1$. □

Lemma 6.2.10 (i) $(?P)P = 1$ in G_2 is reducible to:

(1) $(?F)(\exists A)F = A$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists B)F = B$ in $G_1 * \langle q,q_1\rangle$

(2) $(?F)F = 1$ in $G_1 * \langle q,q_1\rangle$

(ii) G_2 has word problem in $\mathcal{E}^3$.

Proof: (i) 5.1.10.

(ii) Recall from the previous lemma that if $(\exists A)F = A$, then $\text{length}(A) \leq 2\cdot\text{length}(F)$. Also, if $r_i^{-1}A'r_i = B'$, then $\text{length}(B') \leq c\cdot\text{length}(A')$ where c is a constant. (A similar statement holds for "B" in place of "A".) By 5.2.2, the number of steps needed to r-reduce P is in $\mathcal{E}^3$. It remains to show that $G_1 * \langle q,q_1\rangle$ has word problem in $\mathcal{E}^3$. To decide if $F = 1$ in $G_1 * \langle q,q_1\rangle$ we just f-reduce and then see if $\|f[F]\| = 1$, and this procedure is in $\mathcal{E}^3$. □

Corollary 6.2.11 G_1 and G_0 have word problems in $\mathcal{E}^3$.

Proof: G_1 and G_0 are subgroups of a group having word problem in $\mathcal{E}^3$.

□

Lemma 6.2.12 (i) $(?P)(\exists R)P = R$ in G_2 is reducible to: (*)

(1) $(?F)(\exists A)F = A$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists B)F = B$ in $G_1 * \langle q,q_1\rangle$

(2) $(?F)(\exists n)(\exists A)Fx^n = A$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists n)(\exists B)Fx^n = B$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists n)(\exists A)x^nF = A$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists n)(\exists B)x^nF = B$ in $G_1 * \langle q,q_1\rangle$

(3) $(?F)(\exists n)F = x^n$ in $G_1 * \langle q,q_1\rangle$

(ii) (*) is $\mathcal{E}^3$-decidable.

Proof: (i) Given P, to decide if $(\exists R)P = R$ in G_2, first r-reduce P. Let r[P] be the r-reduced word P. There are two cases:

(a) r[P] is r_i-free $(i \in I)$.

Claim: $(\exists R)\ r[P] = R$ in $G_2 \Leftrightarrow (\exists n) r[P] = x^n$ in $G_1 * \langle q,q_1\rangle$.

Proof of claim: $(\Leftarrow)$ Clear.

$(\Rightarrow)$ Suppose $(\exists R)\ r[P] = R$ in G_2. Then $(\exists R$ which is r-reduced$) r[P] = R$ in G_2. By 5.1.11 r[P] and R must contain the same number of r's. Therefore R contains no r's, and so $(\exists n) R = x^n$ in G_2. But,

$$(\exists n) r[P] = x^n \text{ in } G_2 \Leftrightarrow (\exists n) r[P] = x^n \text{ in } G_1 * \langle q,q_1\rangle.$$

(b) r[P] is not r_i-free $(i \in I)$.

If r[P] contains only r's and x's, then we are done. Suppose that the last and first letter of r[P] is not r_i^e $(i \in I,\ e = \pm 1)$ or x^e $(e = \pm 1)$ (else we may consider the word without the final r_i^e or x^e). Then $r[P] = F_1 r_{i_1}^{e_1} P_1 = P_2 r_{i_2}^{e_2} F_2$ where $e_1 = \pm 1$, $e_2 = \pm 1$, and F_1, F_2 are words of $G_1 * \langle q,q_1\rangle$.

(**) If $e_1 = 1$ and $(\exists n)(\exists A) x^n F_1 = A$ in $G_1 * \langle q,q_1\rangle$, then

$$F_1 r_{i_1}^{e_1} P_1 = x^{-n} r_{i_1} r_{i_1}^{-1} x^n F_1 r_{i_1} P_1 = x^{-n} r_{i_1} r_{i_1}^{-1} A r_{i_1} P_1 = x^{-n} r_{i_1} B P_1$$

and so $(\exists R)P = R \Leftrightarrow (\exists R) BP_1 = R$.

If $e_1 = -1$ and $(\exists n)(\exists B) x^n F_1 = B$ in $G_1 * \langle q,q_1\rangle$, then

$$F_1 r_{i_1}^{e_1} P_1 = x^{-n} r_{i_1}^{-1} r_{i_1} x^n F_1 r_{i_1}^{-1} = x^{-n} r_{i_1}^{-1} A P_1$$

and so $(\exists R)P = R \Leftrightarrow (\exists R) AP_1 = R$.

If $e_2 = -1$ and $(\exists n)(\exists A) F_2 x^n = A$ in $G_1 * \langle q,q_1\rangle$, then

$$P_2 r_{i_2}^{e_2} F_2 = P_2 r_{i_2}^{-1} F_2 x^n r_{i_2} r_{i_2}^{-1} x^{-n} = P_2 B r_{i_2}^{-1} x^{-n}$$

and $(\exists R)P = R \Leftrightarrow (\exists R) P_2 B = R$.

If $e_2 = 1$ and $(\exists n)(\exists B)Fx^n = B$ in $G_1 * \langle q,q_1\rangle$, then

$$P_2 r_{i_2}^{e_2} F_2 = P_2 r_{i_2} F_2 x^n r_{i_2}^{-1} r_{i_2} x^{-n} = P_2 A r_{i_2} x^{-n}$$

and $(\exists R)P = R \Leftrightarrow (\exists R)P_2 A = R$.

(***) On the other hand, if $(\exists R)r[P] = R$, then

$$(\exists R)F_1 r_{i_1}^{e_1} P_1 = R = P_2 r_{i_2}^{e_2} F_2$$

i.e., $R^{-1} F_1 r_{i_1}^{e_1} P_1 = 1 \qquad = P_2 r_{i_2}^{e_2} F_2 R^{-1}$.

By Britton's lemma, $R^{-1} F_1 r_{i_1}^{e_1} P_1$ contains a pinch

$$r_{i_1}^{-e_1} x^n F_1 r_{i_1} .$$

If $e_1 = 1$, then $x^n F_1 = A$ in $G_1 * \langle q,q_1\rangle$.

If $e_1 = -1$, then $x^n F_1 = B$ in $G_1 * \langle q,q_1\rangle$.

Similarly for $P_2 r_{i_2}^{e_2} F_2 R^{-1} = 1$.

Thus, for r[P] not r-free, the procedure to decide $(\exists R)P = R$ in G_2 would be as follows:

Decide (using (2))

$$\left.\begin{array}{lll} \text{if } (\exists n)x^n F_1 = A & \text{when} & e_1 = 1 \\ \phantom{\text{if }}(\exists n)x^n F_1 = B & \text{when} & e_1 = -1 \end{array}\right\} \text{ for } P = F_1 r_{i_1}^{e_1} P_1$$

or

$$\left.\begin{array}{lll} \text{if } (\exists n)F_2 x^n = A & \text{when} & e_2 = -1 \\ \phantom{\text{if }}(\exists n)F_2 x^n = B & \text{when} & e_2 = 1 \end{array}\right\} \text{ for } P = P_2 r_{i_2}^{e_2} F_2$$

If not, then by (***) *not* $(\exists R)P = R$.

Else apply the procedure to the resulting word AP_1, BP_1 , P_2B, or P_2A (as given by (**)). On each application of the procedure the number of r's in the word that it is applied to decreases by one. Thus, it can be applied at most a finite number of times.

(ii) By the proof of 6.2.10, the number of steps needed to r-reduce P is f(length(P)), $f \in \mathcal{E}^3$. The length of the resulting r-reduced word is $\leq$ f(length(P)). By 6.2.8, (2) is $\mathcal{E}^3$-decidable. Suppose that to decide (2), g(x) steps are needed for a word of length x (g a nondecreasing function). Then for one application of the procedure for (2), $\leq$ g(f(length(P))) steps are needed. The length of the word to which the procedure for (2) is applied again (as in (**)) is $\leq c \cdot$ f(length(P)) where c is a constant (as in the proof of 6.2.10), and the maximum number of times the procedure for (2) is used is $\leq$ length(P). Therefore the number of steps needed to do (1) and (2) is $\leq \text{length}(P) \cdot g(c^{\text{length}(P)} \cdot f(\text{length}(P)))$. The proof will be completed by showing that $(?F)(\exists n)F = x^n$ in $G_1 * \langle q,q_1\rangle$ is $\mathcal{E}^3$-decidable. It is easily verified that to do this, just f-reduce F, obtaining f[F]; then $(\exists n)F = x^n$ if and only if $\|f[F]\| = x^n$, and this procedure is in $\mathcal{E}^3$.

□

Theorem 6.2.13 G_3 has word problem in $\mathcal{E}^3$.

Proof: 6.2.10 and 6.2.12.

□

Lemma 6.2.14 (i) $(?P)(\exists R_1,R_2)R_1PR_2 = q$ in G_2 is reducible (*) to ·

(1) $(?F)(\exists A)F = A$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists B)F = B$ in $G_1 * \langle q,q_1\rangle$

(2) $(?F)(\exists n)(\exists A)Fx^n = A$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists n)(\exists B)Fx^n = B$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists n)(\exists A)x^nF = A$ in $G_1 * \langle q,q_1\rangle$

$(?F)(\exists n)(\exists B)x^nF = B$ in $G_1 * \langle q,q_1\rangle$

(3) $(?F)(\exists R_1,R_2)R_1FR_2 = q$ in G_2 .

(ii) If (1), (2), and (3) are $\mathcal{E}^n$-decidable, then (*) is $\mathcal{E}$-decidable.

Proof: (i) is almost identical to that of 6.2.12 (i).

(ii) By the proof of 6.1.12 (ii), to reduce (*) to (3) requires $\leq$ h(length(P)) steps, $h \in \mathcal{E}^3$. We will show that (3) is $\mathcal{E}^n$-decidable.

Claim: $(\exists R_1, R_2) R_1 F R_2 = q$ in $G_2 \Leftrightarrow$

$$(\exists m)(\exists n)(\exists X)(\exists Y) F = x^m \bar{X} q_j Y x^n$$

and

$$(\bar{X} q_j Y)^* = q \text{ in } S.$$

Pooof of claim: ($\Leftarrow$) Let $\Sigma = \bar{X} q_j Y$. Then Σ satisfies the hypothesis of Boone's lemma (6.1.2) and so $k^{-1} \Sigma^{-1} t \, \Sigma \, k \, \Sigma^{-1} t^{-1} \Sigma = 1$ in G. Since G is an HNN extension of G_3 with stable letter k, by Britton's lemma, $\Sigma^{-1}\ t\ \Sigma = V$ in G_3 where V is a word on r_i ($i \in I$), x, $q^{-1} t q$. Without loss of generality we can take V to be t-reduced. Then $V \equiv V' R_0 q^{-1} t^e q R_1$. Now,
$\Sigma^{-1}\ t\ \Sigma\ V^{-1} = \Sigma^{-1}\ t\ \Sigma\ R_1^{-1} q^{-1} t^{-e} q R_0^{-1} V'^{-1} = 1$ in G_3.
By Britton's lemma, $\Sigma\ R_1^{-1} q^{-1}$ is a pinch, i.e.,
$\Sigma\ R_1^{-1} q^{-1} = R$ in G_2 , where R is a word on r_i ($i \in I$) and x. Hence $R^{-1}\ \Sigma\ R_1^{-1} = q$ in G_2. Since $F = x^m\ \Sigma\ x^n$, we have the desired result.

($\Rightarrow$) Suppose $(\exists R_1, R_2) R_1 F R_2 = q$ in G_2. (We can take R_1 and R_2 to be r-reduced. Then $R_1^{-1} q R_2^{-1} F^{-1} = 1$ in G_2. By Britton's lemma, $R_1^{-1} q R_2^{-1}$ can be successively r-reduced. But on r-reducing, all the f's to the left of q have negative exponents, and those to the right have positive exponents. Since x's can be moved easily to the right of f_b^e if e = +1 and to the left if e = -1, F has the desired form. To decide if F is of the required form is in $\mathcal{E}^3$: f-reduce F and check if the resulting word is of the desired form by a procedure similar to 6.2.6. If not, then *not* $(\exists R_1, R_2) R_1 F R_2 = q$ in G_2. If so, then since $X q_j Y = q$ in S in $\mathcal{E}^n$-decidable, (3) is $\mathcal{E}^n$-decidable.

□

6.3 $\mathcal{E}^n$-Decidable Groups

Theorem 6.3.1 For $n \geq 4$, there exist finitely presented groups with word problem strictly in $\mathcal{E}^n$. □

Proof: Lemmas of Section 6.2.

If the relation $kq^{-1}tq = q^{-1}tqk$ is omitted in the construction of the groups in the previous sections, then the resulting group has word problem in $\mathcal{E}^3$. Thus, G has a hard "subgroup problem".

Theorem 6.3.2 For $n \geq 2$, there exist finitely presented groups with word problem in $\mathcal{E}^n$. □

Proof: 6.3.1, 5.2.6, 5.2.7.

The "simplest groups", the free groups, have word problem in $\mathcal{E}^2$. It is now known whether the word problem for free groups can be lower than $\mathcal{E}^2$.

CHAPTER 7

OPEN QUESTIONS

(1) Can some of the embedding theorems for groups be shown to be special cases of embedding theorems for logical theories? Sabbagh [1974] notes that the group theoretic analogue of the theorem in logic that a decidable theory has a complete decidable extension is that a group having solvable word problem can be embedded in a simple group having solvable word problem. But there is no connection between the proofs. In the theorem from logic, the resulting extension has the same language as the original theory. In the group embedding theorem, the simple group has additional generators. So to show any connection, it seems as if it is necessary to consider theories extended by adding new symbols. It would be especially interesting to see if the Higman embedding theorem is a special case of a theorem from logic.

(2) Can the word problem for groups be proved unsolvable by constructing a group in which all the generators are functions, similar to the way in which the word problem for semigroups was proved unsolvable by J. Robinson [1968]? This method would have the advantage of presenting a group as a concrete system.

(3) The $\mathcal{E}^n$-decidable groups constructed in this paper were a series of four HNN extensions. Is it possible to construct $\mathcal{E}^n$-decidable groups which are structurally simpler? C. Miller [1971] has shown that the lowest HNN extension having unsolvable word problem is of order two. It may be possible to use his technique in constructing simpler $\mathcal{E}^n$-decidable groups.

BIBLIOGRAPHY

Birkhoff, G., "On the Structure of Abstract Algebras," *Proceedings of the Cambridge Philosophical Society 31* (1935) 433-454.

Boone, W. W., "Word Problems and Recursively Enumerable Degrees of Unsolvability: A Sequel on Finitely Presented Groups," *Annals of Mathematics 84* (1966) 49-84.

Boone, W. W., and Higman, G., "An Algebraic Characterization of Groups with Solvable Word Problem," *Journal of the Australian Mathematical Society 18* (1974) 41-53.

Brainerd, W., and Landweber, L. H., *Theory of Computation*, J. Wiley & Sons (1974), New Y!rk.

Burris, S., "Models in Equational Theories of Unary Algebras," *Algebra Universalis 1* (1971-1972) 386-392.

Cannonito, F., "Hierarchies of Computable Groups and the Word Problem," *J. Symbolic Logic 31* (1966) 376-392.

Clifford, A. H., and Preston, G. B., *The Algebraic Theory of Semigroups, Vols. 1,2*, American Mathematical Society (1967) Providence, R. I.

Cobham, A., "The Intrinsic Computational Difficulty of Functions," *Proceedings of the International Congress on the Logic, Methodology, and the Philosophy of Science* (1964) North Holland (1965), 24-30.

Davis, M., *Computability and Unsolvability*, McGraw-Hill (1958), New York.

Eilenberg, S., and Schutzenberger, M., "On Pseudovarieties," *Advances in Mathematics 19*, No. 3 (1976) 413-418.

Eilenberg, S., and Wright, J. B., "Automata in General Algebras," *Information and Control 11* (1967) 452-470.

Gratzer, G., *Universal Algebra*, Van Nostrand (1968), Princeton, N. J.

Grzegorczyk, A., "Some Classes of Recursive Functions," *Rozpravy Matematyczne iv* (1953) 1-45.

Higman, G., "Finitely Presented Infinite Simple Groups," *Notes on Pure Mathematics 8*, (1974) Department of Pure Mathematics, Australian National University, Canberra.

Higman, G., Neumann, B. H., Neumann, H., "Embedding Theorems for Groups," *J. London Mathematical Society 24* (1949) 249-254.

Hopcroft, J. E., and Ullman, J. D., *Formal Languages and Their Relation to Automata Theory*, Addison-Wesley (1969), Reading, Mass.

Jonsson, B., *Topics in Universal Algebra* (Lecture Notes in Mathematics 250), Springer-Verlag (1972).

Kurosh, A. G., *Theory of Groups*, Chelsea Publishing Company, (1960), New York.

Kurosh, A. G., *Lectures on General Algebra*, Chelsea Publications (1965), New York.

Magnus, W., "Residually Finite Groups," *American Mathematical Society Bulletin 75* (1969) 305-316.

Magnus, W., Karass, A., and Solitar, D., *Combinatorial Group Theory*, Dover Publications (1976), New York.

Meyer, A. R., and Ritchie, D. M., "The Complexity of Loop Programs," *Proceedings of the ACM National Meeting* (1967) 465-469.

McKenzie, R., and Thompson, R. J., "An Elementary Construction of Unsolvable Word Problems in Group Theory," *Word Problems* (Boone, Cannonito, Lyndon, eds.) North Holland (1973).

Miller, C. F., *On Group-Theoretic Decision Problems and Their Classification*, Princeton University Press (1971), Princeton, N. J.

Neumann, B. H., "Some Remarks on Infinite Groups," *J. London Mathematical Society 12* (1937) 120-127.

Neumann, H., *Varities of Groups*, Springer-Verlag (1967), New York.

Oates, S., and Powell, M. B., "Identical Relations in Finite Groups," *J. Algebra 1* (1964) 11-39.

Rabin, M. O., "Recursive Unsolvability of Group Theoretic Problems," *Annals of Mathematics 67*, (1958) 172-194.

Rabin, M. O., "Computable Algebra, General Theory, and Theory of Computable fields," *Transactions AMS 95* (1960) 341-360.

Ritchie, R. W., "Classes of Predictably Computable Functions," *Transactions AMS 106* (1963) 139-173.

Robinson, J., "An Introduction to Hyperarithmetical Functions," *J. Symbolic Logic 3* (1967) 325-342.

Robinson, J., "Recursive Functions of One Variable," *Proceedings AMS 19* (1968) 815-820.

Robinson, R., "Primitive Recursive Functions," *American Mathematical Society Bulletin 53* (1947) 925-942.

Rotman, J. J., *The Theory of Groups*, Allyn and Bacon (1973), Boston.

Sabbagh, G., "Caractérization Algébrique des groupes de type fini ayant un problème de mots résolubles," (Lecture Notes in Mathematics 514), Springer-Verlag (1974) 61-80.

Sacerdote, G. S., "Subgroups of Finitely Presented Groups," *Proceedings London Mathematical Society 35* (Part ii) (1977).

Savitch, W. J., "Relationships between Nondeterministic and Deterministic Tape Complexities," *J. Computer and System Science 4* (1970) 177-192.

Schupp, P. E., "Some Reflections on HNN Extensions," *Proceedings of the Second International Conference on the Theory of Groups*, Canberra (1973) 611-632.

Shoenfield, J., *Mathematical Logic*, Addison Welsey (1967), Reading, Mass.

Tarski, A., "Equational Logic and Equational Theories of Algebras," *Contributions to Mathematical Logic* (Schmidt, Schutte, Thiele, eds.), North Holland (1968) 275-288.

Trahtenbrot, B. A., "Impossibility of an Algorithm for the Decision Problem in Finite Classes," *Doklady Akademii Nauk N.S. 70* (1950) 569-579.